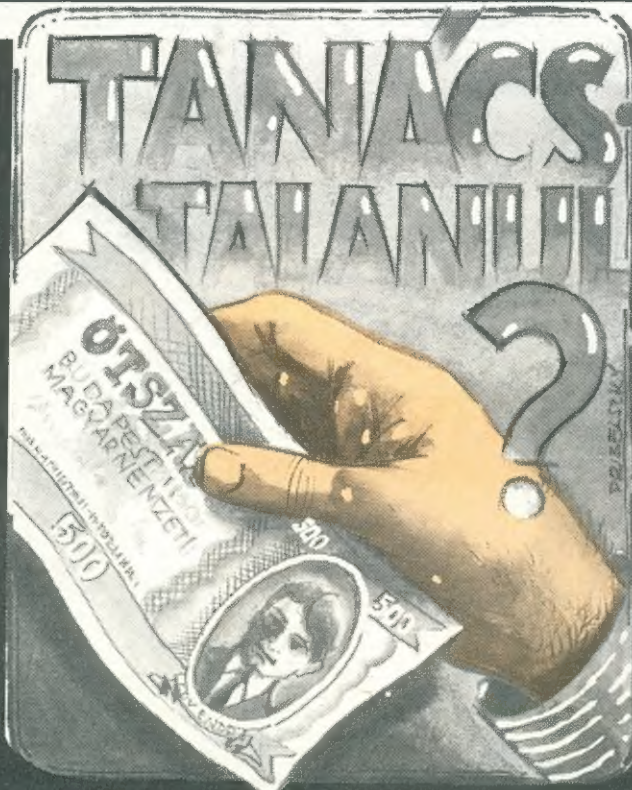


Az elmúlt hónapokban kétségte-  
lenül az IBM-ügy volt  
a számítástechnika világának  
ügye – legalábbis itthon. A sajtó  
legkülönbözőbb fórumain már  
ezerszer lerágták az idevágó  
csontot. Megírták már, hogy  
miért jó, s hogy miért rossz  
a központi pénzeszközök szét-  
osztása a különböző cégek  
között, hogy mennyire mestersé-  
gesek a versenyek illyesfajta  
hazai szabályozásai, hogy ki  
szerint és miért igazságtalanok  
a pénzesztétosztás paraméterei stb.  
Mindezekről tehát nekünk már  
nem érdemes írni, hiszen  
előttünk már mindent leírtak.  
(Lehet persze, hogy nem  
mindent, mi mindenestre nem  
tudunk újabb szempontokat föl-  
sorolni. Maradékalanul ugyan  
nem értünk egyet egyik véle-  
ménnyel sem, de ezzel mégsem  
akarjuk untatni az olvasót.)  
Egész másért hoztuk szóba  
e havi mellékletünk cím-  
oldalán ezt a témát.

Régi mániánk jutott eszünkbe  
az ügy kapcsán. A tanácsadói  
szakma hiánya kis hazánkban.  
Vannak ugyan akik illyesféle  
tanácsadónak nevezik magukat. Az általános tapasztalat  
azonban az, hogy ezek az önjelölt szakemberek még ha nem is  
rossz szakemberek, egy-egy cég elkötelezett hívei, sőt gyak-  
ran munkakönyves vagy csak jól fizetett munkatársai. Mert  
mi lenne, mi a dolga a világ fejlettebb országaiban az ilyen,  
ügynevezett tanácsadónak? Van egy cég, amely elhatározza,  
hogy gépet, gépeket vesz. Tudja, hogy mi a saját cége dolga,  
van is valamilyen elképzelése arról, hogy mindezt, hogyan  
lehetne gépesíteni, nem ismeri azonban – minthogy nem is  
dolga, nem is szakmája – a számítógépes piacot. Fölkeres  
tehát egy független tanácsadói irodát és tanácsot kér. Tan-  
ácsot arra vonatkozóan, hogy az adott cég adott feladatait  
milyen gépekkel lehetne leghatékonyabban és leggazdaságo-  
sabbban végrehajtatni. A tanácsadó természetesen a tanács  
adásakor nemcsak a számításba jöhető gépek tudását, me-  
móriakapacitását, perifériát kell, hogy figyelembe vegye,  
hanem a szükséges szoftverhátteret, azok árát, a különleges  
igényeket stb. Egyszóval nagyon sokféle szempontot.  
Mondhatnák a kedves olvasók, hogy de hiszen az IBM kom-  
patibilis gépeknél ezek a dolgok azonosak. Igaz is meg nem is.



Tudunk olyan IBM kompa-  
tibilis gépről például, amelyen  
még nem láttunk magyar  
ékezetes szövegszerkesztőt,  
holott a gépek többségére  
ilyet már kifejlesztettek.  
A kompatibilitás mértéke mint  
tudjuk különböző. Azután az  
sem mindegy, hogy egy adott  
szoftvert az egyik cégnél  
50 000-ért, egy másiknál egy  
másik kompatibilis gépre  
ugyanazt 100 000-ért kapni.  
A két szoftver között persze  
nem sok a különbség, még az is  
előfordul, hogy mindkettő  
ugyanannak a nyugati termék-  
nek a magyarított, lopott  
változata. Az árak közt azon-  
ban... Ilyen esetben a tanács-  
adó kutya kötelessége lenne  
az ilyen eltérésekre föl hívni  
a vevő figyelmét, tudva azt,  
hogy az adott szoftver a vevő  
érdeklődési körébe  
tartozhat.

Nyilvánvaló tehát, hogy egy  
ilyen tanácsadói iroda, testület  
akkor tölthetné be szerepét,  
ha a gyártóktól, eladóktól  
független lenne.

Márpedig egy ilyen cég föl-  
állításához, működtetéséhez pénz kellene. Pénz és nem is  
keves pénz. (Az már más dolog, hogy megfelelő sikeres mű-  
ködés esetén egy ilyen cég előbb-utóbb nyereséggé is  
válhatna.)

Az az eretnek gondolat jutott az eszembe, hogy ha már ilyen  
tökéletes kondíciókat teremtett a hazai gazdaságirányítás  
néhány cég számára az IBM kompatibilis gépek terjesztéséhez,  
nem kötelezhette volna-e ugyanezeket a cégeket egy ilyen  
tanácsadói szolgálat közös tőkéből való felállítására. Ily-  
módon ugyanis garantálni lehetett volna, hogy a tanácsadó  
cég egyik félnek sem lekötöztettebb a másiknál. Mindezt  
csak feltételes módban írhatjuk le. Tanácsadói iroda létre-  
hozásának eddig ugyanis még a gondolata sem vetődött fel.  
Márpedig a gépet, szoftvert gyártók illetve áru-  
lók közti verseny igazából csak akkor jöhetne létre, ha a potenciális  
vásárlóknak is megadtna az esély a vásárlás előtti reális tá-  
jékozódásra, a kínálatok közti eligazodásra. Ma, szervezett  
tanácsadói apparátus nélkül ez vagy csak álom, vagy ötször  
annyiba kerül a tájékozódni akarónak. S ehhez nem kap hitelt,  
vagy állami támogatást senki.

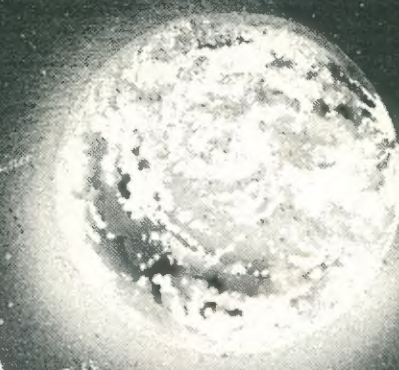
**Angyalosi László**

**BELÜLRŐL**

- 18 **Híroldal** – amelyben megnézhetik maguknak a Commodore Amiga legújabb változatát, a 2000-est
- 20 **Első kézből** a tv computerről – a beépített rutinok közül most a billentyűzetkezelőkkel ismerkedhetnek
- 22 **Programajánlat** – 15-ös játék HT-re
- 23 **Programajánlat** – Primo stopper óriás számjegyekkel
- 24 **Backtrack** – egy programozási módszer rejtelmibe vezetettünk be
- 26 **Mi az a borona?** – azon kívül, hogy egy mezőgazdasági szerszám, megtudhatják
- 27 **Szóljégyzet a BETA BASIC hibáihoz** – a hibák, amelyeket a múlt havi számban közzétettünk, mostanra újabbal egészültek ki!
- 27 **MIKRO MEN** – Hogy mi ez, megtudhatják, ha odalapoznak
- 28 **SORVEZETŐ** – egy régi rovat új kontósben, új tartalommal, változatlan alapkonceptióval – hogy segítsünk a szakköröknek!
- 30 **Könyvmoly** – megrágcsálta az információgazdaságról készült tanulmánykötet mind a 234 lapját
- 31 **Posta** – amelyben egy olvasónknak elmondjuk, hogy mi az a GEOS, mire és hogyan lehet használni
- 31 **Plus/4 nyerő 2. feladat megoldása** – egy nagyon bonyolult feladat megoldása hiányosan. (Ha teljesen elmerülünk benne, sohasem érünk a végére)
- 32 **Atari nyerő** – ez meg egy új pályázat a Skála jóvoltából.



# HIRDAL



## HIPERCHIP

Egy japán szakembercsoport a legnagyobb japán cégek megbízásából már évek óta dolgozik a minden eddigig felülmúló áramköri morzsa, a hiperchip kifejlesztésén. A megvalósulásához közel álló áramkör egy négyszögletes, a mai legnagyobb kapacitású milliméterének kapacitása mintegy hatvan-szorosa lesz a mai legnagyobb kapacitású chipekének. A fejlesztő munka áramkörökből készült — melynek célja ezen áramkörökben az első számítógépek elkészítése — a finanszírozó cégek több mint kétszázhusz a hi-ú eljárást, ismereteket szabadalmaztatott. Ezek egyike révén például elektronsugárral félvezető lapokon — vékonyabb, mint a korábbi lézersugaras módszerrel.

## MONA LISA

Leonardo da Vinci világhírű festményéről, a Mona Lisáról, más néven a Giocondáról újabb érdekes feltételezés született. Lillian Schwartz, számítógéppel dolgozó tanuló kutató közelmúltban közzétett tanulmányában azt állítja, hogy amikor a sejtelsajátmagát Gioconda arcképét megfestette, mes mosolyú Gioconda arcképét megfestette. A kutató egy da Vinci önarcképét elemzése után jutott erre a következtetésre. Lillian Schwartz további feltételezése, hogy Leonardo da Vinci Gioconda-modelljeként egy férfit öltöztetett női ruhába, majd az alak megfestése után az arc ábrázolásakor önarcképét festette meg.

## MIKROSKÓP

Fizikai Nobel-díjjal jutalmazták a legújabb elven működő elektronmikroszkóp kifejlesztőit. Az STM (pásztorzó alagúthatású mikroszkóp) segítségével először nyílt lehetőség megfigyelni az atomokat összetartó elektron-kötéseket. Az STM-ben a lencsék egy rendkívül érzékeny, vékony tűvel kialakított érzékelője helyettesíti. A tű mindössze néhány angstrom távolságban halad végig a vizsgált felület felett. Az alkalmazott villamos feszültség olyan áramot kelt, melynek nagysága a tű és a felület közötti távolságtól függ. A képszűrés úgy emeli vagy süllyeszti a tűt, hogy az áram értéke, illetve a tű és a felület közötti távolság azonos maradjon. Így válik lehetővé, hogy a tű folyamatosan mozgatva egy számítógép segítségével a vizsgált felület atomi finomságú, háromdimenziós képe legyen látható.

## GYORSÍTÓ

Vannak olyan cégek is, amelyek nem új géptípusok fejlesztésén, hanem meglévő gyártmányok feljavításán dolgoznak. Így például az amerikai PCSG cég olyan speciális áramköri kártyát hozott forgalomba, amellyel jelentősen gyorsítható az IBM PC működése. A gyártó cég szerint a mindössze háromszáz dollárért forgalmazott kártyával az IBM PC tulajdonosok gépeket a jóval magasabb szintű IBM PC-AT kategóriába emelhetik.

## KAYPRO 386

Az amerikai Kaypro Corporation a közelmúltban jelentette be a 16 MHz-es, Intel 80386-os mikroprocesszorra épülő, három változatban készülő számítógépét, a Kaypro 386-ot. Az alapmodell 512K-s RAM-ot és 1,2 Megabyte-os hajlékonylemez meghajtót foglal magába. A géphez EGA típusú monochrome képernyő tartozik. A Kaypro 386 tárkapacitása jelentősen bővíthető: 660 Megabyte-ig merevlemezre és 16 Megabyte-ig RAM-ig.

## KUANTUNG

Kína Kuantung tartományában tervezik felépíteni az amerikai IBM első kínai mikroszámítógépgyártó üzemét. Az első sorban professzionális személyi számítógépeket gyártó üzemben a kínai és az ázsiai piacra gyártanak. A most folyó tárgyalásokon szó van arról, hogy kezdetben csak összeszerelés folyjon Kuantungban, majd később pedig már kínai alkatrészeket is beépítenének a gépekbe. Az IBM néhány éven belül mintegy kétszázmillió dolláros kínai forgalomra számolt.

## ULTRAHANGOS

Új, ultrahangos vizsgálati módszert vezettek be osztrák orvosok a fertőző májgyulladás megállapítására. A betegséggyanús májszövetéről ultrahangos felvételt készítenek és azt számítógép segítségével összehasonlítják egy beteg máj felvételével. A módszer fölősslegessé teszi a hosszadalmas vérvizsgálatot. Hasonló számítógépes kiértékelési eljárást rosszindulatú daganatok ultrahangos felvételeinek elemzésére is alkalmazzanak.

## EQUITY III +

Az Epson America cég legújabb IBM PC-AT kompatibilis gépe 80286 típusú processzorral épült, választhatóan 6,8 és 10 MHz-es órásebességgel. Az egyes sebességek a válaszadási frontidő alatt lévő kapcsolóval választhatók. Az Equity III+ 640 Kbyte-os memóriával rendelkezik, 64 Kbyte-os ROM-jal, 128 Kbyte-os bővíthető és MS-DOS 3.2 operációs rendszer alatt fut. Az új Epson gép 1,2 Mbyte-os floppy meghajtót tartalmaz és felszerelhető hozzá: színes, grafikus adapter EGA kártya, színes és fekete/fehér monitorok.

## FELVÉTELI

Januárban tettek felvételi vizsgát a szeptembertől induló, két tannyelvű képzést nyújtó gimnáziumokban jelentkező tanulók. Hét gimnázium összesen tizenkét osztályba hároméves nyolcadik feladatlapokat töltött ki, amelyekkel logikai, matematikai gondolkodásukat, kreativitásukat vizsgálták. A feladatlap kitöltése után a felvételizők válaszait számítógépbe táplálták, értékelték és rangsorolták.



**PUMA**

**PUMA**

Az NSZK-beli, világhírű sportruházati cég rászolgált a márka becsületére. Legújabb, különleges megjelentetése RS-Electronic elvezetésű futócipője. A már sorozatban gyártott futócipő kész elektronikai csoda: memóriát, mikroprocesszort, nyomásérzékelőt, csatlóegységet stb. tartalmaz. Mindehhez a vásárló még szoftvert és csatlókábelt is kap. A csodapipő futás közben méri az időt, az egyes lépések számát, a futó szervezete által felhasznált energiát. A levett futócipőhöz csatlakoztatható és a mért adatok különféle szempontok szerint feldolgozhatók. Első sorban az edzői és a rehabilitációs munkában számítanak az elektronikus Puma cipő sikerére.

## MIKROPERIFÉRIA

## MIKROPERIFÉRIA

Nemcsak a mikroszámítógépek teljesítménye nő napról napra az új fejlesztési eredmények kapcsán, hanem a mikroszámítógép perifériái is egyre gyorsabbak, egyre nagyobb kapacitásúak lesznek. E folyamatra jellemző a Konica cég új floppy egysége, amely az általános 1 Mbyte-al szemben 10 Mbyte-kapacitású.

**SZINGAPÚR**

**SZINGAPÚR**

A félvezető és mikroelektronikai iparáról egyre ismertebbé és híresebbé váló Szingapúr rendkívül olcsó áraiival is magára vonja a világ figyelmét. Részben az ottani alacsonyabb munkabérek, részben a legsikeresebb műszaki konstrukciók lemásolása révén megta- karított fejlesztési költségek és nem utolsó sorban a piaci versenyszándék következté- ben olcsón kínálják elektronikai cikkeit. Például egy háromszázötven darab 256 K-s chippel szerelt PC kártya ára mindössze ezer dollárba kerül. Nyolc-kilencszáz dollárért már XT kategóriájú, IBM kompatibilis pro- fesszionális PC-k vásárolhatók a szigetor- szágból. S mindehhez tulajdonképpen egy üres floppy lemez árért is meg lehet kapni. alkalmazási szoftvereket is meg lehet kapni.

## GYŐZTESEK

## GYŐZTESEK

Hét Nyugat-európai mikroszámítógépes szaklap a múlt évben meghirdette az év szoftverje versenyt. A lapok olvasói négy kategóriában szavaztak. Az egyes kategóriák győztesei a következők voltak. A kereskedelmi tárgyú programok közül első a rendkívül könnyen kezelhető, és az adatokat grafikon is megjeleníteni tudó Javelin lett, megelőzve a Supercalc 4.0, a d'Basse III+, és Lotus 1-2-3 programokat. A technikai-tudományos kategóriában a grafikai lehetőségeiben kiemelkedő Autocad győzött. Jellemző a gyors rajzkészég, háromdimenziós grafika, ábrák tengely körüli forgatása, részletkicsinyítés és nagyítás, stb. A szoftverkétesítőket versenyében a könnyen kezelhető rendkívül gyors programírást biztosító Turbo Pascal verte a mezőnyt. S végül a szórakoztató programok kategóriájában meglepetésként Arthur Dent, egy csupán szöveges információkat kiíró, de szellemes, izgalmas sci-fi történetre épülő játék programja lett a 86-os év első helyezettje.

**KORNYEZETVEDELEM**

# KÖRNYEZETVEDELMI

Ipari környezetvédelmi információs rendszert épít ki az Ipari Minisztérium. Az ipar felhalmozódott környezetvédelmi gondjainak megoldásához pontos információkra van szükség. Ezért még az idén számítógépes adatbankot hoznak létre, amelybe betáplálják többek között az Országos Vízügyi Hivatal, a Meteorológiai Intézet és a Köjál már meglévő környezetvédelmi információit, s az iparvállalatoknál is megszervezik a legfontosabb környezetvédelmi adatok gyűjtését. A számítógépes adatbankot az egy éve működő Ipari Környezetvédelmi Koordinációs Iroda fogja üzemeltetni. A szakemberek már hozzáláttak a számítógépes adatbankját építéséhez, az Ipari Minisztérium Vízügyi Hivatal hasonló rendeltetésű számítógépes rendszerével, s a későbbiekben más hálózatokkal is összeköttetést létesítenek. Az új információs rendszer jelentős segítséget nyújt a környezetvédelmi szakembereknek a veszélyforrások feltárásához és folyamatos figyéléséhez. Az adatbankban tárolt információkat jól hasznosíthatják a környezetvédelmi fejlesztések előkészítésekor. Az információs rendszer még az idén megkezdí működését.

# AMIGA 2000

**AMIGA 2000**

A Commodore cég legújabb gépe a hazai számítógépes körökben is nagy sikert aratott AMIGA 2000-es változata. Kisöccse az 1000-es ott volt decemberi BIT-LET Karácsonyunkon is. Az új Amiga megőrizte elődje kiváló grafikai és zenei képességeit, s emellett kompatibilis az IBM PC-vel! Bővítő kártyával pedig XT-vé varázsolható. A gépben három meghajtó van, két 3,5 collos és egy 5,25 collos. Ráadásul mindegyik meghajtót lehet használni IBM üzemmódban is, meg Amiga üzemmódban is.

Memóriaterülete 1 Megabyte. Workbench 1.2-es operációs rendszerrel dolgozik, illetve PC XT üzemmódban az MS-DOS operációs rendszert használja. Processzorai: Motorola 68000, valamint Agnus - grafikai és animációs-chip, - Desi - video-chip - és Paula - interface-chip.

19





## ELSŐ KÉZBŐL

A TV COMPUTER RÓL

## BILLENTYŰZET RUTINOK

A beépített rutinok leírását a billentyűzetkezelővel folytatjuk. A billentyűzet, valamint a beépített és a külső botkormányok leolvasását végzik. A használt rendszerváltozók a következők:

**PICTURE** 10 byte, címe 2897=0B51h

Az automatikus billentyűzetismétlés kezdetéhez szükséges idő 20 ms-os egységben. BASIC-ből a SET DELAY állítja.

**OLDPICTURE** 10 byte, címe 2907=0B5Bh

A billentyűzetmátrix előzőleg leolvasott értéke

**DELAY KEY** 1 byte, címe 2917=0B65h

Az automatikus billentyűzetismétlés kezdetéhez szükséges idő 20 ms-os egységben. BASIC-ből a SET DELAY állítja.

**LOCK KEY** 1 byte, címe 2918=0B66h

Az aktuális kurzor (LOCK) állapotát mutatja:

0 normál karakterek (LOCK, normál kurzor)

1 nagybetűk (LOCK+CTRL, inverz C kurzor)

2 folyamatos shift (LOCK+SHIFT, inverz S kurzor)

8 alternatív karakterek (LOCK+ALT, inverz A kurzor)

**RATE KEY** 1 byte, címe 2919=0B67h

Automatikus billentyűismétlés alatt a két karakter megjelenése előtti időt adja 20 ms-os egységben. BASIC-ből a SET RATE állítja.

**HOLD DIS** 1 byte, címe 2920=0B68h

Jelzi, hogy a CTRL+P együttes lenyomásával felfüggeszthető-e a rendszer működése. (Pl. listázás félbeszakítható)

0 HOLD-mód engedélyezve

255 HOLD-mód tiltva

### A rutinok leírása:

**KBD IRQ** hívási kód: 144 (90h)

működés: A billentyűzetmátrix leolvasását végzi.

Felhasználói programból nem ajánlott hívni.

**KBD CHIN** hívási kód: 145 (91h)

output: C=karakter vagy botkormány kódja

A=hibakód

működés: A billentyűzeten leütött karakter, vagy a botkormány elmozdításának kódját adja. Megvárja, amíg érkezik kód.

**KBD STATUS** hívási kód: 147 (93h)

output: C=jelző

0 nincs beolvasható karakter

255 beolvasható karakter

működés: A billentyűzet, ill. botkormányok státusát adja

### A billentyűzet közvetlen leolvasása

A billentyűzet, ill. botkormányok közvetlen leolvasására is szükség lehet egyes programokban.

Az 1. táblázat megadja a billentyűzetmátrixot. Ennek leolvasásához először ki kell választani a mátrix megfelelő sorát (0-9), és ezt a 3. portra az alsó 4 biten kiküldeni. A beolvasás az 58h portról történik. Az a billentyű volt éppen lenyomva, amelyikhez tartozó bit értéke zérus.

A 3. port egyéb bitjeinek is van jelentése, ezért a sorkiválasztást az utoljára kiküldött byte alsó 4 bitje helyébe kell írni, és úgy kiküldeni.

A megfelelő rendszerváltozó:

**PORT03** 1 byte, címe 2833=0B11h

bit7, bit6: bővítőkártya kiválasztás

bit3-bit0: billentyűzetmátrix sorkiválasztás

Pl. ahhoz, hogy a RETURN billentyű le van-e nyomva, az 1. táblázat alapján az 5. sor 4. bitjét kell vizsgálni. Ezt a vizsgálatot végzi el az 1. program gépi kódú rutinja. Ha visszatéréskor a ZERO FLAG=0, akkor igen, egyébként pedig nem volt a RETURN lenyomva.

Cseh Tibor

1. táblázat: Billentyűzet-mátrix a TV-Computeren.

sor	7.	6.	5.	4.	3.	2.	1.	0.
0	!	'	i	/	&	"	+	%
1	=	ö	ó	#	ü	)	(	~
2	R	Q	@	Z	;	W	E	T
3	U	P	ü	{	ö	o	i	}
4	F	A	<	H	\	S	D	G
5	J	é	ő	RET	Á	L	K	DEL
6	V	Y	LOCK	N	SH	X	C	B
7	M	-	SP	CTRL	ESC	:	?	ALT
8		RJL	RJR	RJA	RJF	RJD	RJU	INS
9		LJL	LJR	LJA	LJF	LJD	LJU	

A rövidítések jelentése:

RET : RETURN

SP : szóköz

SH : SHIFT

\* RJL : balra

\* RJR : jobbra

\* RJA : gyorsítás vagy 2.tűz

\* RJF : tűz

\* RJD : le

\* RJU : föl

beépített (\*) és  
jobboldali külső  
botkormány

LJL : balra

LJR : jobbra

LJA : gyorsítás vagy 2.tűz

LJF : tűz

LJD : le

LJU : föl

baloldali külső  
botkormány

```

10 !
20 !-----!
30 !
40 !          TV-COMPUTER          !
50 !
60 ! RETURN billentyű figyelése    !
70 ! a billentyűzet-mátrixon      !
80 !
90 !-----!
100 !
110 ! A gépi kódú rutin:
120 !
130 !          =0B11      PORT03      EQU      0B11h
140 !
150 ! 243          F3          DI
160 ! 58,17,11      3A 11 0B      LD      A,(PORT03)
170 ! 230,240      E6 F0          AND      0F0h
180 ! 246,5         F6 05          OR       5
190 ! 50,17,11      32 11 0B      LD      (PORT03),A
200 ! 211,3         D3 03          OUT     (3),A
210 ! 219,88       DB 58          IN      A,(58h)
220 ! 203,103      CB 67          BIT     4,A
230 ! 251          FB          EI
240 ! 201          C9          RET
250 !

```



## Egy hasznos segédprogram:

## APPEND

Korábban már ismertettük a MERGE-eljárást programok összefűzésére. Az ott bemutatott módon tetszőleges programot be tudunk illeszteni a memóriában levő programunkba. Csupán egy apró kényelmetlenséggel jár a dolog: a beszúrandó programból lista file-t kell készíteni, és ezt kell szalagról leolvasni. Némi programozási „áldozattal” gyorsabb és egyszerűbb megoldást biztosít a 2. program. A listából csupán az 1. és 2. sort kell begépelni, a többi sor csak a működés könnyebb megértése miatt szerepel.

Az említett programozási áldozat a következő: a programokat úgy kell megírni, hogy az egyes részeknek különböző tartományba essenek a sorszámai. Pl. a főprogram sorszámai 100–999 tartományba esnek, a hozzá tartozó külön megírt szubrutinok pl. az 1000–1999, 2000–2999 stb. tartományt használják. Lényeges, hogy ne legyen átfedés! A használt legkisebb sorszáma 3 lehet, mert az első két sorban van az APPEND programunk. Ezek után jöhet az összefűzés,

sorszámok szerint növekvő sorrendben!

Először betöltjük az APPEND programot és elindítjuk:

RUN

Az 1. ábra szerinti képet fogjuk látni, a kurzor a 3. sor elején villog. Két RETURN után a kurzor a 7. sorban levő LOAD elején villog. Ha szükséges, beírhatjuk a LOAD mögé idézőjelek között a betöltendő program nevét. Újabb RETURN-re a gép kiírja:

Searching

Most kell a legkisebb sorszámkokat tartalmazó programrészletet betölteni. Legyen ez pl. a „FÖPROGRAM” nevű. Ha a betöltés kész (2. ábra), nyomjunk meg néhány RETURN-t, amíg az alul levő sorokat is végrehajtja a BASIC (3. ábra). Most újabb RUN paranccsal kezdve ismétlés, amíg növekvő sorszámok szerint az utolsó programrészletet is be nem olvastuk. Ezután kitöröljük az APPEND programot:

DELETE-2

és kimenthetjük SAVE paranccsal az összefűzött egész programot.

### A program működési elve

a 2. program listájában szereplő rövid gépi kódú program megkeresi az aktuális BASIC programunk végét. A BASIC program kezdőcímét a TEXT rendszerváltozó tartalmazza (címe 5922=1722h). Az első két POKE utasítás átállítja a TEXT változót, így az a programunk végére fog mutatni. Ennyi azonban nem elég a betöltés előtt, mert a LOAD vagy NEW utasítás hatására a TEXT változóba beíródik a BASIC számára használható legkisebb cím. Ezt állítja ugyancsak a program végére a második két POKE. Az említett rendszerváltozó a VLOMEM (címe 5920=1720h). Ezek után a BASIC már „nem látja” az APPEND programot, a betöltött új program fizikailag mégis ennek folytatásaként kerül a memóriába. A harmadik két POKE visszaállítja VLOMEM-et, a negyedik kettő pedig a TEXT-et az eredeti programkezdetre. Újabb RUN-ra az eljárás ismételhető.

Cs. T.

### 2. program

```

1 A$="":FORI=1TO13:READB:A$=A$&CHR$(B):NEXT:T=USR(2+VARPTR(A$
)):TH=INT(T/256):TL=T-256*TH:RCL=PEEK(5922):RCH=PEEK(5923):GRAPH
ICS4:PRINT"*Betöltés előtt:"PRINT:PRINT"poke5922,";TL;":poke592
3,";TH:PRINT:PRINT"poke5920,";TL;":poke5921,";TH
2 PRINT:PRINT"load":PRINTAT18,1:"*Betöltés után:"PRINT"poke5
920,";RCL;":poke5921,";RCH:PRINT:PRINT"poke5922,";RCL;":poke5923
,";RCH:PRINTAT2,1:"":END:DATA42,34,23,126,167,200,133,111,48,24
9,36,24,246
10 !--- Csak az 1. és 2. sort kell begépelni ---!
20 !-----!
30 !-----!
40 !          TV-COMPUTER          !
50 !-----!
60 ! BASIC programok összefűzése !
70 !          ( APPEND )          !
80 !-----!
90 !-----!
100 !
110 ! A használt gépi kódú rutin:
120 !
130 !          =1722      TEXT      EQU      1722h      ;program eleje
140 !
150 ! 42,34,23      2A 22 17      LD      HL,(TEXT)
160 !                      LOOP:
170 ! 126              7E              LD      A,(HL)
180 ! 167              A7              AND      A
190 ! 200              C8              RET      Z      ;program vége
200 ! 133              85              ADD      A,L
210 ! 111              6F              LD      L,A
220 ! 48,249          30 F9              JR      NZ,LOOP
230 ! 36              24              INC      H
240 ! 24,246          18 F6              JR      LOOP
250 !

```

```

*Betöltés előtt:
ok
poke5922, 97 :poke5923, 27
poke5920, 97 :poke5921, 27
load

```

```

*Betöltés után:
poke5920, 239 :poke5921, 25
poke5922, 239 :poke5923, 25

```

1. ábra

```

*Betöltés előtt:
ok
poke5922, 97 :poke5923, 27
ok
poke5920, 97 :poke5921, 27
ok
load

```

```

Searching
Reading: FÖPROGRAM
ok

```

```

*Betöltés után:
poke5920, 239 :poke5921, 25
poke5922, 239 :poke5923, 25

```

2. ábra

```

*Betöltés előtt:
ok
poke5922, 97 :poke5923, 27
ok
poke5920, 97 :poke5921, 27
ok
load

```

```

Searching
Reading: FÖPROGRAM
ok

```

```

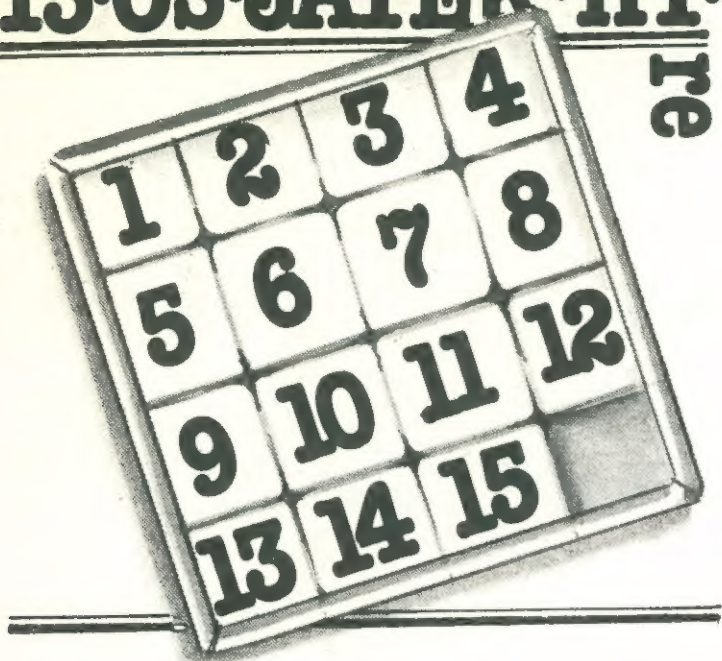
*Betöltés után:
poke5920, 239 :poke5921, 25
ok
poke5922, 239 :poke5923, 25
ok

```

3. ábra



# 15-ÖS JÁTÉK HT.



Egy ősrégi játék. Megvalósítása programozástechnikailag is szellemes. A játék maga pedig máig érdekes.

Használatához való információkat megadja a program.

## Felépítése:

- 50 Keret
- 60-80 0-9 a nagyméretű számok előállítására  
A\$(0)-A\$(9)-be.  
(Azt használja föl, hogy a számok csak 9 féle karakterből állnak.)
- 90 Ugyanez 10-15-ig.
- 100-110 A számok elé szókötők, A(X,0)=az X szám helye a képernyőn A(X,1)=az A(X,0) helyen lévő szám.  
D\$-ban a jó sorrend.
- 130 Összekeverés
- 140 Főprogram. Benyomott betűt hasonlítja nyílakkal, ettől függően a négy irányító rutinra ugrás. Ha B\$ egyenlő D\$-ral, akkor össze van rakva GOTO 230  
(B\$ az állásnak megfelelően változik az irányító rutinokban.)

150-220 Irányító rutinok.

230-240 "ÚJRA?"

Móricz Zsigmond Gimnázium szakköre - Tiszakécske

```

10 '+++++
15 '++      15-ÖS JATEK      ++
20 '++ KESZITETTE: PETROV FERENC ++
30 '+++++
40 CLEAR 1000:DEFINT A-Z:DIMA$(16),A(16,1):CLS
50 FORX=25TO92:SET(X,13):SET(X,40):NEXT:FORX=13TO40:SET(24,X):SET(25,X):SET(92,X)
  :SET(93,X):NEXT
60 A$="128131143179191140188176":FORX=1TOLEN(A$)STEP3:B$(X)=(X-1)/3:CHR$(VAL(MID$(A$,X,3))):NEXT
70 A$="656474076004556433556334606114655334600434556004656434656114":FORX=0TO9:F
  DRY=1TO6:A$(X)=A$(X)+B$(VAL(MID$(A$,X*6+Y,1))):IFY=3THENA$(X)=A$(X)+CHR$(26)+STR
  ING$(3,24)
80 NEXT Y,X
90 FORX=10TO15:A$(X)=A$(1)+" "+CHR$(27)+A$(VAL(RIGHT$(STR$(X),1))):NEXT
100 FORX=1TO15:A$(X)=A$(X)+" ":IFX<10THENA$(X)=" "+A$(X)
110 Y=325+FIX(X/4-.1)*128+(X-FIX(X/4-.1)*4)*8:PRINTY,A$(X):A(X,0)=Y:A(X,1)=X:B
  $(X)=B$+CHR$(X):NEXT:B$=B$+CHR$(16):D$=B$
120 A$(16)=" "+CHR$(26)+STRING$(8,24)+" ":A=16:A(A,0)=741:X=16
130 PRINT0,"OSSZEKEVEREM! ":FORU=1TO200:ONRND(4)GOSUB170,150,210,190:NEXT:PRINT
  0,"SZAMOK BETOLASA AZ URES HELYRE A NYILAKKAL!"
140 A$=INKEY$:ON ABS(A$="A")+A$(CHR$(10))*2+A$(CHR$(8))*3+A$(CHR$(9))*4GOSU
  B170,150,210,190:IFB$=D$THEN230ELSE140
150 IFX>4THENA(X,1)=A(X-4,1):PRINTA(X-4,0),A$(16):PRINTA(X,0),A$(A(X-4,1)):B
  $(X)=LEFT$(B$,X-5)+CHR$(16)+MID$(B$,X-3,3)+MID$(B$,X-4,1)+RIGHT$(B$,16-X):X=X-4
160 RETURN
170 IF X<13THENA(X,1)=A(X+4,1):PRINTA(X+4,0),A$(16):PRINTA(X,0),A$(A(X+4,1)):
  B$=LEFT$(B$,X-1)+MID$(B$,X+4,1)+MID$(B$,X+1,3)+CHR$(16)+RIGHT$(B$,12-X):X=X+4
180 RETURN
190 IF(X-1)/4<INT((X-1)/4)THENA(X,1)=A(X-1,1):PRINTA(X-1,0),A$(16):PRINTA(X,
  0),A$(A(X-1,1)):B$=LEFT$(B$,X-2)+CHR$(16)+MID$(B$,X-1,1)+RIGHT$(B$,16-X):X=X-1
200 RETURN
210 IF(X/4<INT(X/4))THENA(X,1)=A(X+1,1):PRINTA(X+1,0),A$(16):PRINTA(X,0),A$(
  A(X+1,1)):B$=LEFT$(B$,X-1)+MID$(B$,X+1,1)+CHR$(16)+RIGHT$(B$,15-X):X=X+1
220 RETURN
230 PRINT0,"VEGRE SIKERULT OSSZERAKNOD!"":PRINT64,"JATSZOL ME
  G EGYET? (I)?"
240 A$=INKEY$:IFA$="I"THENRUNELSEIFA$=""THEN240
  
```



# PROGRAMMA. IÁNLAT:

**PRIMO**  
stopper óriás számjegyekkel

Közkedvelt programozási feladat a stopper készítése – alapváltozatát már a kezdők is kipróbálják. A most közölt megoldás magasabb igényeket is kielégít: pontosabb a szokásos BASIC-időzítésnél, a számjegyek pedig a normál karakter sokszorosai, megjelenítésük a lehető leggyorsabb képváltással történik. Ezt természetesen gépi kódú szubrutinokkal oldottam meg, amelyek a maguk nemében nagyon egyszerűek. Tanulók számára éppen arra lehetnek bátorító példák, hogy látszólag bonyolult feladatok is meglepően leegyszerűsíthetők. Az időmérő (tulajdonképpen késleltető) szubrutin tized másodpercet mér. Mivel a kijelzés egész másodperces, erre csak azért van szükség, hogy a leállító billentyűnyomásra azonnal reagáljon a gép, teljesebb legyen az illúzió. A szubrutin assembler alakja:

```
LD DE, 20CFH
DEC DE
LD A,D
OR E
JR NZ, FBH
RET
```

Az időmérés finom szabályozása a CFH (207) érték módosításával végezhető el. A számjegyek rajzát a következőképpen állítjuk elő. A ROM 12791-es címétől kezdve található a karakterek kódolt mintázata 8-8 byte-on (az A típusjelű gépeken a 30-as, a B jelűeken a 26-os ASCII kódtól kezdve készült a táblázat). Bonyolult és lassú lenne minden egyes számjegyrajzoláshoz innen dekódolni és megjeleníteni. Ehelyett csak egyszer végeztetjük el a dekódolást a program előkészítő részében: a számjegyek adatait célszerű formában átmásoltatjuk egy szabad tárolóterületre. Minden számjegy 7x5 pontos mátrixban jelenik meg – az új tárolás viszont úgy történik, hogy a számjegyek adatai 35 byte-on helyezkedjenek el (az eredeti 0-s bit 0-s értékű byte-ot, az 1-es bit 255-ös értékű byte-ot hoz létre). Az átkódolást a 3110-től 3150-ig terjedő programrész végzi.

```
LD HL, az a
cim, ahová
másolni kell
LD A, 05H
PUSH AF
PUSH HL
LD C, 07H
LD A, /DE/
PUSH DE
LD DE, 0020H
LD B, 0CH
LD /HL/, A
ADD HL, DE
DJNZ FCH
POP DE
INC DE
DEC C
JR NZ, F0H
POP HL
INC HL
POP AF
DEC A
JR NZ, E6H
RET
```

A számjegyek váltásakor ezt a 35 byte-ot másolja át egy gépi kódú szubrutin a képernyőre 12-szeres nagyítással, így a méret végül 40x84 képernyőpont lesz. A másolóprogram assembler listája: A program közvetlenül vesz át adatokat a ROM-ból, ezért gondoskodni kell az A és B típus felismeréséről. Ennek egyik lehetséges megoldása a 3030-as sor. Arra is szükség van, hogy a program alkalmazkodjék az eltérő memóriaméretű gépekhez, mert a gépi kódú rajzolónak ismernie kell a képernyő-RAM címét. Ezt a PEEK(20) közvetlenül jelzi.

A módszertani tanulságokon túl jó szolgálatot is tehet a program: nagyobb közönség előtt rendezett vetélkedőkön, játékokon időmérésre használhatjuk.

**Fekete György** 7300 Komló, Bocskai u. 35.

```
10 REM *****
20 REM * P R I M O - S T O P P E R *
30 REM *****
40 GOSUB 3010: REM ELOKESZITES, FELIRAT
100 REM MUKODTETES
110 Z=INKEY$: IF Z="$" THEN 310
120 IF Z<>"*" THEN 110 ELSE BEEP 20,20
130 FOR A=0 TO 5:POKE G,J:E=CALL(F,A(A))
140 FOR B=0 TO 9:POKE G,K:E=CALL(F,A(B))
150 FOR C=0 TO 5:POKE G,L:E=CALL(F,A(C))
160 FOR D=0 TO 9:POKE G,M:E=CALL(F,A(D))
170 BEEP 1,1: FOR N=0 TO 9
180 E=CALL(P): IF INKEY$="-" THEN 210
190 NEXT: NEXT: NEXT: NEXT: NEXT
200 REM MEGALLITAS, FOLYTATAS
210 BEEP 200,100
220 Z=INKEY$
230 IF Z="0" THEN GOSUB 1010: GOTO 110
240 IF Z="*" THEN BEEP 20,20: GOTO 190
250 IF Z<>"$" THEN 220
300 REM BEFEJEZES
310 POKE R,PEEK(R)+128: OUT 0,PEEK(R)
320 POKE T,24: CLS: END
999 REM *****
1000 REM NULLAZAS
1010 POKE G,J: GOSUB 2010: BEEP 500,10
1020 POKE G,K: GOSUB 2010: BEEP 400,12
1030 POKE G,L: GOSUB 2010: BEEP 300,14
1040 POKE G,M: GOSUB 2010: BEEP 200,16
1050 FOR A=0 TO 10: BEEP 10,40
1060 BEEP 40,10: NEXT: RETURN
2000 REM A NULLA IRATASA
2010 E=CALL(F,A(0)): RETURN
3000 REM FELIRAT, ELOKESZITES, ADATOK
3010 DEFINT A-T: DEFSTR Y,Z: CLS
3020 F=16476: P=17000: C=P+10
3030 D=12935: IF PEEK(D)=128 THEN E=4
3040 G=F+1: B=D+E*8: A=PEEK(20): S=A-1
3050 Y="S T O P P E R": Z=CHR$(228-E)
3060 PRINT$ 1,8,CHR$(2)CHR$(5)YCHR$(21)
3070 PRINT$ 7,20,Z: PRINT$ 9,20,Z:
3080 PRINT$ 12,2,"*: INDUL -: MEGALL"
3090 PRINT$ 13,2,"$: VEGE 0: NULLAZ"
3100 PRINT$ 14,16,"*: FOLYTATJA"CHR$(1)
3110 K=255: FOR D=0 TO 9: A(D)=C
3120 FOR E=6 TO 2 STEP -1: H=2^E
3130 BEEP D,E: FOR N=1 TO 7: M=PEEK(B+N)
3140 IFM AND H THEN POKEC,K ELSE POKEC,0
3150 C=C+1: NEXT: NEXT: B=B+8: NEXT
3160 POKE P,17,207,32,27,122,179,32,
251,201: REM KESLELTETO (1/10 SEC)
3170 POKE F,33,1,A+6,62,5,245,229,14,7,
26,213,17,32,0,6,12,119,25,16,252,
209,19,13,32,240,225,35,241,61,32,
230,201: REM SZAMJEGYRAJZOLO
3180 J=1: K=9: L=18: M=26: P=16443
3190 POKE R,PEEK(R) AND 127: T=R+11
3200 POKE T,1: GOSUB 1010: RETURN
```



# egy programozási módszer:

# BACKTRACK

A személyi számítógépek rohamos elterjedésével egyre több diák és felnőtt jut gépközelbe az iskolában, családban vagy a munkahelyen, és lázasan tanul programozni. Ez többnyire azt jelenti, hogy ismerkedik az adott gép BASIC utasításaival, arra törekedve, hogy minél több és furfangosabb utasítás váljon ismertté számára. Mint ahogy nem állíthatjuk, hogy valaki tud sakkozni, ha ismeri a játék szabályait, ugyanúgy nem biztos, hogy „tud” programozni az, aki elég sok BASIC utasítást ismer.

Néhány példán keresztül megpróbálunk bemutatni egy programozási módszert, amely többnyire nem igényel bonyolult utasításokat, csak némi ötletességet. A mellékelt programok Commodore-16-os gépre készültek, de könnyen átirhatók bármely más géptípusra is.

Az általános iskolai tanulók számára kiírt egyéni pályázat matematikai feladatai között szerepel az alábbi:

5 radírt, 4 ceruzát és 3 tollat vásároltunk összesen 100 forintért. 1 radír kevesebbe, 1 toll pedig többbe kerül, mint 1 ceruza. Mindegyiknek egész forint az ára. Mennyibe kerülhet külön-külön 1 radír, 1 ceruza és 1 toll? (1985/86. tanév, I. forduló, 3. feladat.)

Legyen a radír ára  $x$ , a ceruzáé  $y$ , a tollé  $z$ . A feladat szerint olyan  $x$ ,  $y$ ,  $z$  számhármast kell keresnünk, amelyre  $5x+4y+3z=100$ , ahol  $x$ ,  $y$ ,  $z$  pozitív egész, és  $x < y < z$ .

Először ki fogunk választani egy megfelelő  $x$  értéket, ehhez egy megfelelő  $y$ -t, végül  $z$ -t.

$x$  kiválasztásakor 1-gyel kezdve minden pozitív egész szám szóba jöhet addig, amíg nem lesz az  $5x+4(y+1)+3(x+2)$  kifejezés értéke 100-nál nagyobb. (Itt használtuk ki az  $x < y < z$  feltételt.)

Valamelyik  $x$  értéket kiválasztva lényegében elindultunk egy „úton”, amelyen  $y$  kiválasztása újabb „ételágazást” jelent, ahol a továbbhaladáshoz ismét választunk egy „utat”. Itt az elágazások száma már  $x$  értékétől is függ.  $y$  lehetséges értékei az  $x$ -nél nagyobb, és az  $5x+4y+3(y+1) \leq 100$  feltételnek eleget tevő pozitív egész számok.

$z$ -t is  $y+1$ -gyel kezdve növeljük egyesével addig, amíg  $5x+4y+3z$  éppen 100 lesz – azaz megkapunk egy megoldást –, vagy meghaladja a 100-at, ekkor *visszalépünk*, és vesszük a következő  $y$  értéket. Ha egy adott  $x$  értékhez tartozó minden  $y$  értéket meg-

vizsgáltunk, akkor ismét *visszalépünk* (még egyvel alacsonyabb szintre), és vesszük a következő lehetséges  $x$  értéket, hozzá  $y$ -t majd  $z$ -t stb.

## Mindezt így realizálhatjuk egy programban:

```
10 SCNCLR
20 X=0
30 X=X+1
40 IF 5*X+4*(X+1)+3*(X+2)>100 THEN END
50 Y=X
60 Y=Y+1
70 IF 5*X+4*Y+3*(Y+1)>100 THEN 30
80 Z=Y
90 Z=Z+1
100 IF 5*X+4*Y+3*Z<100 THEN 90
110 IF 5*X+4*Y+3*Z>100 THEN 60
120 PRINTX,Y,Z
130 GOTO 60
```

Figyeljük meg, hogy ez a rövid program milyen kevés BASIC utasítást használ. Kétségkívül fel lehetne – sőt fel is kell – „cicomázni” a programot (áttekinthetőbb kiírás, megoldások számolása stb.), de ez a program lényegét nem érinti.

A probléma megoldásának természetesen nem ez az egyetlen útja. Pl.  $x$  és  $y$  ismeretében az  $5x+4y+3z=100$  egyenletből  $z$  már kiszámítható, de ekkor azt kellene vizsgálnunk, hogy  $z$  egész szám-e.

## Az egész csokoládénak

mekkora részét ehette meg Ági, Béla, Cili és Dénes?

```
10 REM 1/A+1/B+1/C+1/D=1 MEGOLDASAI
20 M$="".##. 1/# + 1/# + 1/## + 1/## = 1"
30 SCNCLR
40 A=1
50 A=A+1:IF A>4 THEN END
60 B=A-1
70 B=B+1
80 IF B+A>=A*B THEN 70
90 IF B+3*A<A*B THEN 50
100 C=B-1
110 C=C+1
120 IF B*C+A*C+A*B>=A*B*C THEN 110
130 IF B*C+A*C+2*A*B<A*B*C THEN 70
140 D=C-1
150 D=D+1
160 IF B*C*D+A*C*D+A*B*D+A*B*C>=A*B*C*D THEN 150
170 IF B*C*D+A*C*D+A*B*D+A*D<A*B*C*D THEN 110
180 I=I+1:PRINTUSING M$;I,A,B,C,D
190 GOTO 110
```

Ebben a feladatban is pozitív egész számokat keresünk, melyekre teljesülnek az

$$\frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d} = 1 \text{ és } a \leq b \leq c \leq d$$

feltételek.

A *visszalépésről* kapta a módszer a nevét. (Backtrack = visszalép, visszatancol.) Ehelyett nevezhetnénk ezt az eljárást módszeres próbálgatásnak is.

Ugyancsak általános iskolai matematikai versenyfeladat (egyéni pályázat 1984/85. tanév, II. forduló, A/4 feladat) a következő:

Ági, Béla, Cili és Dénes testvérek. Születésük sorrendje megegyezik a leírt sorrenddel. Egy tábla csokit úgy osztottak el egymás között, hogy egyik sem evett többet, mint a nála fiatalabb. Mindegyikük csokoládéja az egész csokinak olyan törttel kifejezhető része, amelyiknek a számlálója 1 és a nevezője egész szám.

(Ági csokoládérése  $\frac{1}{d}$ , Béláé  $\frac{1}{c}$  stb.)

A mellékelt program ugyancsak a backtrack elvén működik.

Pl.  $c$  legelső kipróbált értéke  $b$ . Ha még  $\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \geq 1$ , akkor  $c$ -t tovább kell nö-

velnünk, de ha már  $\frac{1}{a} + \frac{1}{b} + \frac{1}{c} < 1$ ,



```

10 REM ** FELIRAT, TABLA **
20 W1$=" "
30 W2$=" "
40 W3$=" "
50 W4$=" "
60 W5$=" A B C D E F G H "
70 LIST0,6,0:LIST4,6,0:LIST1,2,7:CLR
80 SAVE1,8,6,"NYOLC VEZER A SAKKTABLAN"
90 SAVE1,0,22,"AZ L-T MEGNYOMVA LEPESENK
ENT HALADUNK."
100 T0=200+T1
110 IF T1<T0 THEN 110:GOTO CLR
120 SAVE1,10,4,W1$
130 FOR W=1 TO 8:W$=CHR$(57-W)+" "+W2$
140 SAVE1,8,3+2*W,W$
150 SAVE1,10,4+2*W,W3$
160 NEXT W
170 SAVE1,10,20,W4$
180 SAVE 1,10,22,W5$
190 REM***** START *****
200 DIM A(8,8),S(8):I=0
210 REM ** VEGYUK A KOVETKEZO OSZLOPOT *
*
220 I=I+1:J=0
230 REM ** VEGYUK A KOVETKEZO MEZOT**
240 J=J+1
250 REM ** ELERTUK AZ OSZLOP VEGET? **
260 IF J>8 THEN 380
270 REM ** FOGLALT A MEZO? **
280 IF A(I,J) > 0 THEN 240
290 REM ** LETESZ EGY JELET **
300 A$=" " : E=1 : S(I)=J : GOSUB 430
310 REM ** VAN UJ OSZLOP? **
320 IF I<8 THEN 220
330 REM ** MEGOLDAST TALAL **
340 Q=Q+1:PRINT"Q",Q: " MEGOLDAS"
350 L$="L":GOSUB500
360 PRINT"Q"
=9
370 REM ** VISSZ AZ ELOZO OSZLOPRA **
380 I=I-1:IF I=0 THEN 530
390 REM ** FELVESZ EGY JELET **
400 A$=" " : E=-1 : J=S(I) : GOSUB430
410 IF I=8 THEN 380:GOTO 240
420 REM ** MEGJELOLT - ADMINISZTRALT **
430 SAVE1,9+2*I,21-2*J,A$
440 FOR X=I TO 8
450 A(X,J)=A(X,J)+E
460 J1=J+X-1:J2=J-X+1
470 IF J1<9 THEN A(X,J1)=A(X,J1)+E
480 IF J2>0 THEN A(X,J2)=A(X,J2)+E
490 NEXT X
500 IF L$="L" THEN GET/ L$
510 IF L$<>"L" THEN GETL$
520 RETURN
530 SAVE1,27,2," V E G E ! "
540 END

```

akkor c-vel túlhaladtunk a megfelelő értéken, így vissza kell lépnünk b növelésére. Ezeket a feltételeket azonban nem vizsgálhatjuk ebben a formában, közös nevezőre hozással át kell írunk őket úgy, hogy a vizsgált egyenlőtlenségekben egész számok szerepeljenek.

Próbáljuk átírni a programot arra az esetre, ha öt testvérnek kell osztoznia a csokoládén, ugyanilyen feltételek mellett.

Ebből már kitűnne a backtrack módszer hátránya is. Ha ugyanis nagyon sok esetet kell megvizsgálni, akkor ez még a gép számára is sokáig tarthat. Pl. öt testvér esetén az első megoldás

$$\frac{1}{2} + \frac{1}{3} + \frac{1}{7} + \frac{1}{43} + \frac{1}{1806} = 1,$$

és ez több mint egy perc futásidő után derül ki.

Sokat javíthatunk a program sebességén, ha úgy írjuk meg a programot, hogy a gép mielőbb ismerje fel, hogy a vizsgált út nem hoz eredményt. Pl. itt az első négy szám ismeretében ki lehet számolni, hogy az ötödiket milyen szám közelében kell keresnünk. Ezzel viszont bonyolultabbá válik a program.

A backtrack módszer illusztrálására klasszikusnak számító feladat a következő:

Helyezzünk el a sakktáblán nyolc vezért úgy, hogy azok ne üssék egymást. Keressük meg az összes megoldást.

Figyeljük meg a mellékelt programban, hogy egy figura letevésekor vagy visszavételekor ugyanaz a szubrutin használható (420—520), csak míg a letevéskor (300) minden foglalttá váló mező „súlyát” 1-gyel növeljük, addig a felvételkor (400) 1-gyel csökkentjük. Így a program számontartja, hogy az adott mező hányszorosan foglalt. Ha ez a szám 0, akkor szabad letenni egy figurát.

Végezetül álljon itt egy probléma, melynek a megoldásához jól használható a backtrack módszer:

Válasszunk ki az 1 és n közötti természetes számok közül minél többet úgy, hogy bármely kettőnek a különbsége ne legyen egyenlő bármely másik kettő különbségével. Pl. n=12 esetén kiválasztható öt ilyen szám: 1, 2, 5, 10, 12. Van-e másik öt ilyen szám? Ki lehet-e választani hatot? Ha nem, milyen n esetén választható ki hat (hét, nyolc stb.) ilyen szám?

A probléma megoldásához sok sikert kíván

**Szilassi Lajos** Szeged, Bite Pál u. 2/b

A cikk példái talán nem tűntek eléggé meggyőzőnek. Bizonyára van olyan olvasó, aki ezek alapján nem látta be a backtrack módszer hasznosságát.

Igaz, hogy az első példában bemutatott egyenlőségfajta megoldására létezik hatékony algoritmus. Gyorsabb, mint a backtrack – de ha ezt használjuk, a számítógépes program már korántsem lesz ilyen egyszerű és áttekinthető. Igaz az is, hogy egy kis előzetes gondolkodás után a második feladat is megoldható ügyesebben; azonban ezt is csak a program egyszerűségének rovására tehetjük meg.

A backtrack lényegét igazán a harmadik példa mutatja meg. Ezt a feladatot valóban nem lehet hatékonyabban megoldani. A módszer ilyen jellegű elrendezési problémáknál használható igazán – legyen szó akár hasonló logikai feladatról vagy egy sakktáblán elemzéséről, akár egy iskola órarendjének elkészítéséről, akár pedig egy sokváltozós, egymást befolyásoló tényezők szerint működő rendszer modellezéséről. Ezekben az esetekben egyedül a backtrack ad belátható időn belül használható eredményt.

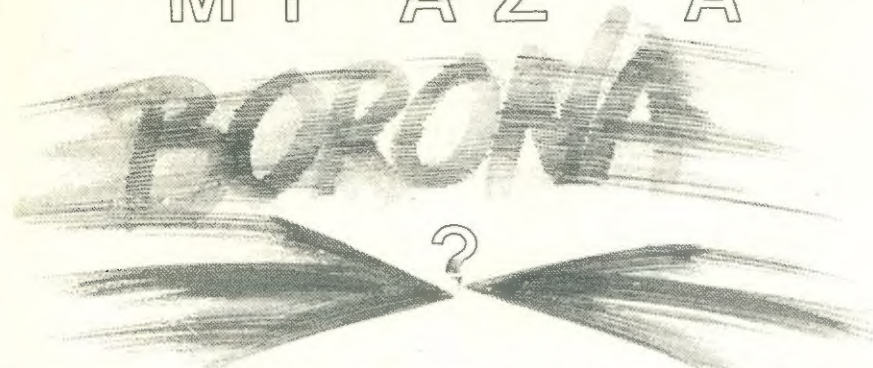
**A szerkesztő azért van,**

**hogy a lap olyan legyen,**

**amilyenek az olvasói!**



# M I A Z A



**Peremkerület kis mellékutcájának egyik épületén tábla: Csokonai Művelődési Ház. A BIT-LET olvasói számára nyilván nem ismeretlen a név, hiszen itt tartottuk az első BIT-LET Karácsonyt. Ez a ház a Borona székhelye.**

Nem mezőgazdasági kiállítás nyílt itt, a Borona a számítógépes cserepartner- és márkatárs-kereső szolgálat „fedőneve”.

– Úgy indult az egész – mondja Tóth Lajos igazgató –, hogy négy évvel ezelőtt mi kezdtünk el elsőként foglalkozni a számítástechnikával a művelődési házak között. Jó szoftveres és hardveres szakembereket sikerült összegyűjtenünk, így klubjaink rövid idő alatt népszerűvé váltak, sőt tanfolyamokat is tudtunk szervezni.

Valóban, a Csokonai mikrogépes klubjának jelenleg mintegy kilencven tagja van, ezen felül a csak itt működő Sinclair OL klub tagjainak száma is hatvan. Így arra kényszerültek, hogy „arisztokratizálják” a klubtagságot: ma már csak az léphet be, akit valamelyik régebbi klubtag ajánl.

– Erre a szakembergárdára építve hoztuk létre a Mikroinform nevű szolgáltatásunkat. Arra vállalkoztunk, hogy bárkinek, aki akár személyesen, akár levélben hozzánk fordul a mikroszámítógéppel kapcsolatos kérdésekkel, annak ingyenesen megadjuk, illetve megküldjük a választ. A Mikroinform bővült ki később, a BIT-LET Karácsony idején a Boronával. Ennek keretében az azonos érdeklődési körű számítógéptulajdonosok teremthetnek kapcsolatot egymással. Aki mikrogépes levelező- vagy cseretársat keres, az egy kitöltött kérdőív és ötven forint befizetése ellenében megkapja az azonos érdeklődési körű partnerek nevét és címét.

Az igény erre hatalmas, hiszen a kis településeken élőknek szinte semmiféle lehetőségük nincs arra, hogy megtalálják érdeklődési körüknek megfelelő társaikat – ezt bizonyítja a szerkesztőségünkhöz érkező rengeteg levél is. Így a művelődési házat a legkisebb falvakból is naponta keresik.

– Arra, hogy sokan mennyire fontosnak tartják a kapcsolatteremtést, jó példa az, amikor az egyik „társkereső” címét hibásan küldtük meg az érdeklődő partnereknek. Nemcsak ő telefonált a hiba javítását kérve, hanem még négyen-ötven, az ország legkülönbözőbb pontjairól, akiknek a rossz címzés ellenére sikerült felvenniük a kapcsolatot a márkatárssal. Mára már oda jutottunk, hogy kénytelenek vagyunk szűkíteni, pontos-

sabbá tenni az érdeklődési köröket, mert nem tudjuk például egy-egy Commodore-tulajdonosnak valamennyi márkatársa címét elküldeni. A napokban készül el az új kérdőívünk, amely az előzőnél sokkal jobban behatárolja azt, hogy kit milyen jellegű programok érdekelnek. Arra már nem vállalkozhatunk, hogy annak is közvetítsünk partnert, akit „minden” érdekel.

Mint hogy a tanácsadó szolgálat ingyenes, a Borona pedig mindössze ötven forintba kerül, a vállalkozás nem tűnik nyereségesnek – pedig az. Nyereséges, hiszen az információ is pénzt ér. A Csokonai olyan adatbázisra tett szert szolgáltatói tevékenysége révén, ami kamatozik. Közel 1500 mikrogéptulajdonos adatait tartják nyilván, és olyan tekintélyt vívtak ki, hogy vállalatok, intézmények egymás után hívják őket tanfolyamok megtartására, és sok az önkéntes felajánlás, anyagi támogatás is – így az Ipari Informatikai Központ ötvenezer forint értékű szakkönyvet adott át a háznak.

– Annyira megnőtt az adatállományunk, hogy most keresünk egy olyan céget, mely hosszú távra kölcsönözne nekünk egy komolyabb gépet a nyilvántartási munkák megkönnyítésére. Ezért cserében a szétküldött leveleinkben propagál-nánk is a gépet adó intézményt.

A Csokonai Művelődési Házban nemrégiben megszűnt a szokásos heti disco is. Táncolni kevesebben akartak, mint számítógépet kezelni, így győzött a mikrogépesek akarata, akiket zavart a hangos zene. A számítógépesek fanatikusak, így elérték azt is, hogy a házban a hét minden napján van számítógépes rendezvény.

– Hozzánk bárki, bármikor bejöhet, és használhatja a gépeket – feltéve, hogy programozni vagy tanulni akar rajtuk, és nem játszani. Ahhoz nincs elegendő számítógépünk, hogy a gyerekek ezeken vívjanak ürhaborút. Szervezünk családi tanfolyamokat is, érdekes módon itt jóval kisebb a lemorzsolódás, mint a csak felnőtteknek szóló tanfolyamainknál.

A magyarázat könnyen kitalálható. Ha valaki munka mellett akarja elsajátítani a programozás alapfogásait, hamar úgy érzi, hogy kevés erre az ideje, és fárasztja az újszerű tanulnivaló. Ha azonban azt látja, hogy a fia vele együtt halad, akkor szívesebben fogadja, ha lemaradna tőle.

– Szombatonként csaiádias a hangulat: felnőttek, gyerekek együtt dolgoznak. 9-től 14 óráig minden géptulajdonosnak óránként 20 forintért biztosítjuk az áramot és a monitorot. Ha valaki magával hozza a tv-t, akkor ingyen jöhet.

A nagy januári havazás idején a házban minden program elmaradt. Kiürült a ház, az emberek inkább otthon maradtak a fűtött szobában. Csak szombat délelőtt dideregetek néhányan a kapu előtt a hóviharban, hónuk alatt számítógéppel, spárgával összekötött monitorral.

– Valódi közösség alakult ki itt. Egyszer az egyik klubtagunknak, egy tizenéves fiúnak eltűnt a gépe. A többi tag gyűjtést szervezett, és negyed óra alatt összeadták a pénzt – így a fiú a következő héten már újra dolgozhatott az új gépen. A számítógépesek megszállottak, sőt kicsit őrültek – de nem lehet nem szeretni őket – fejezi be Tóth Lajos. **Tallér József**

**A Borona olyan vállalkozás, amelyet nemcsak a Csokonai Művelődési Ház-zal kiépített régi jó kapcsolatunk miatt tartunk reklámozásra érdemesnek. Úgy gondoljuk ugyanis, hogy nemcsak a Művelődési Ház, de a hobbi-számítógépesek, köztük olvasóink érdekeit is szolgálja.**

**A Borona tehát elérhető az alábbi címen: Csokonai Művelődési Ház – 1153 Budapest, Eötvös u. 64–66. Telefon: 690-495, 892-240**

**KERAVILL MEV**

**µELEKTRONIKAI**

**MÁRKABOLT**

BP.V., MŰZEUM Krt.11.

**MIKROELEKTRONIKA:**

**A JÖVŐ A JELENBEN.**

\*\*\*\*\*

**FELVEZETŐK,**

**INTEGRÁLT ÁRAMKÖRÖK,**

**MIKROPROCESSZOROK**

**ÉS CSATLAKOZÓK.**

SZAKTANÁCSADÁS, CSOMAGKÜLDŐ SZOLGÁLAT.



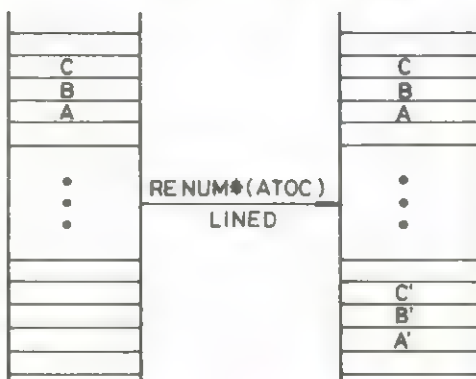
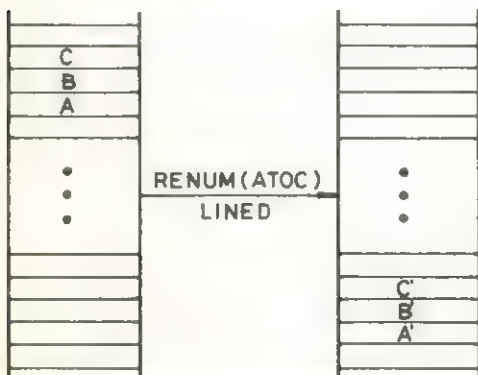
# A BETA BASIC

**Januári számunkban beszámoltunk a BETA BASIC bővítés néhány hibájáról, amelyek a DELETE és a KEYIN utasításokkal kapcsolatosak. Azóta kiderült, hogy a BETA BASIC még kevésbé tökéletes, mint eredetileg hittük. Most egy újabb hibára hívjuk fel a figyelmet.**

Szinte minden toolkit és BASIC bővítés parancslistájában megtaláljuk a RENUMBER utasítást. Ez a programsorok átszámozását hivatott elvégezni.

Az utasítás működésének megértéséhez először nézzünk meg két másik parancsot. Az egyik a MOVE, a másik a COPY. A MOVE egy vagy több programsor átmozgatását végzi el, vagyis az eredeti sorok törölődnek, és újabbak írődnek be a kiválasztott helyre. A COPY az adott sort, illetve sorokat a megfelelő területre másolja, de ekkor az eredeti sorok nem törölődnek.

A BETA BASIC 3.0 változatban a MOVE utasításnak gyakorlatilag a RENUM (re-number, azaz újraszámozás) felel meg, a COPY-nak pedig a RENUM\*. A RENUM és a RENUM\* működését, különbözőségeiket az ábra mutatja:



A vessző nélküli és a vesszővel jelzett programsorok tartalma megegyezik, csak sor-számaik térnek el egymástól.

Mindez hasznos szolgáltatás, de van a RENUM® utasításnak egy súlyos fogyaté-  
kossága. Ez akkor jelentkezik, ha csak egyet-  
len sort kívánunk átmásolni, és a másolni  
kívánt sor, valamint a célsorszám között  
nincsenek további programsorok.

Nézzünk erre egy mintapéldát. Gépeljük be:

10 REM

és adjuk ki a  
RENUM\* (10 TO 10) LINE 100  
parancsot. Ha lenyomjuk az ENTER-t, a  
következő programlistát kapjuk:

10 REM  
100 REM  
10 REM

Vagyis az átmásolt sor az eredeti sorszámaival együtt a célsorszám mögött is megjelenik. Ha ezután a kurzort lefelé mozgatjuk a képernyőn, akkor alul értelmetlen sorrendben jelennek meg a „bűnös” sorok. Óvatosan kísérletezzünk, mert a többi kurzormozgató billentyű lenyomására el is szállhat a rendszer!

Ha újra lenyomjuk az ENTER-t, akkor a listában már két kurzor is látható. A felesleges sor csak az előtte lévő sor (illetve sorok) eltávolítása után tüntethető el, de ezzel együtt persze először is az eredeti sort kell törölni.

A hibát az okozza, hogy az utasítás értelmezésekor rosszul működik a TO utáni ellenőrző rutin. Ha az átmásolás közvetlenül az áthelyezni kívánt sor utáni területre történik, akkor az eredeti sort az értelmező egy két sorból álló programblokknak tekinti, így a célsor után megismétli. Emiatt lesz két azonos számú sorunk. Ha a start- és a célsor között van még programsor, akkor a rutin helyesen hajtja végre feladatát.

A BETA BASIC szerzői nem vették észre a hibát, hiszen az utasítás jelentősége első-sorban nagyobb programblokkok átmásolásánál érezhető.

Egyetlen sor másolása a legegyszerűbben úgy valósítható meg hiba nélkül, ha a start-sort lehívjuk EDIT-tel, átírjuk a sorszámát, és ENTER-rel ismét bevisszük a memóriába.

**Rucz Lajos**

**Programcsereborjú szoftverbörze bemutatók és vásárlás**

a Miskolci Nehézipari Műszaki Egyetem Könyvtárában  
1987. április 10–11.

### A Programcserebere teremben asztalbérlet

tévé és csatlakozási lehetőség	– óránként 20 forint
ugyanez Commodore 64-el és magnóval	– óránként 50 forint
ugyanez Commodore 64-el és floppyval	– óránként 60 forint

**Előjegyzésben mindez 20% kedvezménnyel.** Előjegyzés telefonon az Egyetem KISZ Bizottságánál: 06-46-65-111/11-04

Az előjegyzett helyeket négy napig tartják, ez idő alatt kell befizetni

**a pénzt a koveckező elmre:**

**MIKRO MEN – KISZ Bizottság – 3515 Miskolc** – Egyetemváros  
Lapzártakor a programok szervezése még tart. Lesznek program- és gép-  
bemutatók, vállalatok, felhasználók találkozhatnak IBM kompatibilis gépekre  
írott szoftverekkel és azok szerzőivel. Lesz bemutató tanítás a számítástech-  
nikát oktató tanároknak, találkozó az egyetem szakembereivel a fiatal köz-  
gazdászok és műszakiak részére.

**A miskolci egyetemi könyvtár ezen a két napon csak a számítógépeseké!**

**Belépő: diákoknak és katonáknak 10 forint, másoknak 20 forint**

**Nyitva: reggel 9-től este 7-ig**





**Szubjektív véleményem szerint a BIT-LET egyik legpozitívabb rovata a SORVEZETŐ volt. Több írásával vitatkoztam, de olyan „hézagpótló” szerepet játszott, ami nagyobb hibákkal szemben is elnézővé tett volna. Konkrétabban: a szinte teljesen elhanyagolt tanárképzés területén próbálkozott... Rossz ómen ennek a hamvába-holt sorozatnak a felélesztését célul tűzni, de feltétlenül szükség lenne valami hasonlóra, ezért megpróbálkozunk vele.**

1986 végén komoly központi támogatással indult egy kísérlet „A számítástechnika matematika orientált oktatása” címmel. Különböző adminisztratív okok miatt a kísérlet óvatos csendben indult el két irányba: Simonovits Miklós vezeti az ún. IRÁNYÍTOTT vonalat. Ennek keretén belül 8–10 középiskolában olyan tanterv szerint oktatnak, amelyik S. M. „Számítástechnika” tankönyvének alapul. A másik vonalon 10–12 pedagógus saját elképzeléseit próbálgatja. Új Sorvezetőkben megpróbáljuk erről a kísérletről röviden informálni a BIT-LET-et olvasó tanárokat. Szeretnénk például hasznos anyagokat közölni nem teljesen kezdő foglalkozásokhoz (szakköri és órai munkához egyaránt). Ezenkívül bárki ötletének, módszereinek, észrevételeinek szívesen biztosítunk fórumot.

## A kísérletről röviden

A Művelődésügyi Minisztérium Köznevelési Kutatások Titkársága által menedzselte kísérlet célja egy olyan koncepció kialakítása, melyben a számítástechnikai ismeretek közvetlen egységben jelentkeznek a társ-tantárgyakkal, esetünkben a matematikával. Nem állítjuk, hogy a matematika mással nem helyettesíthető, pusztán arról van szó, hogy a kísérlet vezetői és résztvevői első sorban ehhez értenek és vonzódnak. Kapcsolatra törekszünk más irányokkal is.

Részletesebben azt szeretnénk, hogy a matematikaórán tanult számítógépes ismeretek egy sokkal eredményesebb, alaposabb, és magasabb szintű matematika tanításhoz vezessenek. Mindenkor átlagos – nem tagozatos – osztályokat veszünk célba, még ha ennek eleinte ellent is mond a kísérletben részt vevő iskolák színvonala. Nem szakköri körülményekben gondolkodunk, de próbálunk tekintettel lenni a diákok különböző szintű alapismereteire.

A résztvevő pedagógusok egy néhány oldalas tematikát adtak be: hogyan képzelik el az oktatást. Tanév közben kéthavonta küldenek óravázlatokat, illetve naplót: mikor, mi történt valójában. Év végén pár oldalon foglalják össze, mit, hogyan sikerült megvalósítani, mik a tanulságok. Évente 2–3 alkalommal összegyűlve megismerhetik egymás elképzeléseit, gondjait, és tervezünk szakmai továbbképzést, bemutatót is.

Részben a kísérlethez, (részben hagyományainkhoz) kapcsolódik egy évközi haladó szakkör, amelyet több középiskolával közösen többek között a KFKI nyári táborainak leghaladóbb tagjainak tartunk. Ennek anyagából valók az alábbi feladatok. Külön föl hívjuk az olvasók figyelmét a színes alányomással kiemelt kérdésekre. Ezek megoldásai beküldhetők. Ebből egyrészt az anyagok visszhangjára szeretnénk következtetni, másrészt sok kérdésre magunk sem ismerjük az egzakt feleletet. A helyes (és frappáns) válaszok beküldői között apróságokat (floppy, kazetta, BIT-LET-ek stb.) sorsolunk ki. **A megoldásokat** kivételesen nem a BIT-LET hanem a szerző címére küldjük. **Azaz: Török Turul – 1525 Budapest MTA KFKI Pf.: 49.**

## Bolyongástól a Lissajoux görbékig

**1. feladat.** Tekintsünk a képernyőn egy A, B oldalú téglalapot. Ennek egy belső pontjából kiindulva determinisztikusan „bolyong” egy pont. Azaz, az oldalakkal 45°-os szöget bezáró egyenesek mentén mozoghat, és a határokon rugalmasan visszaverődik. Használjunk „karakter-grafikát” (CHAR, PRINT, PRINT AT stb.), és A, B illetve a kezdőpont legyen véletlenszerű!

```
20 DEFFNR(X)=INT(RND(1)*X+1)
40 X1=FNR(30)+8:Y1=FNR(15)+8
45 X=FNR(X1-1):Y=FNR(Y1-1)
46 SDCNCLR
60 FOR I=0 TO X1:CHAR 1,I,Y1+1,"":NEXT
70 FOR I=0 TO Y1:CHAR 1,X1+1,I,"":NEXT
80 DX=1:DY=1
90 FOR I=1 TO 2*X1*Y1
100 X=X+DX:Y=Y+DY
110 IF ABS(X-X1/2)>X1/2-.1 THEN DX=-DX
115 IF ABS(Y-Y1/2)>Y1/2-.1 THEN DY=-DY
120 CHAR 1,X,Y,"•"
130 NEXT I
150 GETKEY A$:CHAR 1,5,24,STR$(X1)+STR$(Y1)
```

X1 és Y1 az oldalhosszak, 45-ben X és Y a kezdőpont koordinátái. DX és DY (mindkettő 1 vagy -1) a bolyongás pillanatnyi irányát határozzák meg.

Futtassuk néhányszor a programot! Remélhetőleg mindenkiben kérdések és észrevételek tömege fogalmazódik meg, pld.:

a) Mikor van vége a „dolognak”? Mi történik amikor nem történik semmi?

b) Lehetne-e gyorsítani a bolyongást, hiszen nagyobb téglalapon bizony unalmas egy kicsit.

c) Mitől függ az ábra telítettsége? Próbáljunk valami törvényszerűséget felfedezni!

Gondolkozzunk el egy kicsit a kérdéseken. A gyorsítás lényegében technikai probléma: vegyük észre, hogy az IF-es sorok (110–115) túl gyakran kerülnek végrehajtásra. Valóban felesleges minden lépésnél megnézni, fálnál vagyunk-e. Előre ki is számíthatjuk, mikor kell visszaverődni, pontosabban: vízszintes, avagy függőleges fal lesz a következő. Némi töprengés után adódik az alábbi változat:

```
5 DEFFNR(X)=INT(RND(1)*X+1)
7 A=FNR(100)+20:B=FNR(100)+20
10 VX(2)=A+5:VY(2)=B+5:VX(0)=5:VY(0)=5
20 X=FNR(10)+5:Y=FNR(15)+5:DX=1:DY=1:XX=X:YY=Y
100 GRAPHIC 4,1
105 CHAR 1,5,19,STR$(A)+STR$(B)
110 BOX 1,VX(0),VY(0),VX(2),VY(2):DRAW 1
150 X1=X:Y1=Y:V1=ABS(X-VX(DX+1)):V2=ABS(Y-VY(DY+1))
160 IF V1<V2 THEN X=X+V1*DX:Y=Y+V1*DY:OR
170 IF V2<V1 THEN X=X+V2*DX:Y=Y+V2*DY:DY
=-DY:DRAW 1 TO X,Y:GO TO 150
180 X=X+V1*DX:Y=Y+V2*DY:DRAW 1 TO X,Y
190 DX=-DX:DY=-DY:GO TO 150
```

**Három eset van:**

– függőleges fal – 160. sor

– vízszintes fal – 170. sor

– mindkettő egyszerre (sarok) – 180–190. sorok

X1 és Y1 jelentése is megváltozott, a legutóbbi visszaverődés helyét mutatják. Ez a program már méltó a finom grafikára. A további gyorsítás már összefügg az első kérdéssel. Figyeljük meg, hogy bizonyos esetekben egy időre megszakad a bolyongás, majd a kiindulási pontból ellenkező irányba folytatódik. Mikor lesz ilyen, és valóban „várokozik”-e ilyenkor a gép?



## Két esetet kell megkülönböztetnünk:

– A mozgó pont eljut valamelyik sarokba. Ilyenkor úgy verődik vissza, hogy saját addigi nyomán halad (vissza) a kiindulásig, majd onnan ellenkező irányba szalad. Hol végződik ilyenkor a bolyongás: melyik lesz az utolsónak kivilágított pont?

– Nem megy sarokba a pont. Hogyan végződik ilyenkor a bolyongás? Mitől függ, a két eset közül melyik következik be?

**Próbáljuk úgy módosítani a programot, hogy ne dolgozzon hiába! Minél kevesebb ponton haladjunk át többször! Ez igen sokféleképpen oldható meg, mutatunk egy lehetőséget, de reméljük az olvasóktól kapunk szellemesebbet is! (Az 5–150-es sorok változatlanul átvethetők az előző programból!)**

```
160 IF V1<V2 THEN X=X+V1*DX:Y=Y+V1*DY:DR
AW 1 TO X,Y:DX=-DX:GO TO 200
170 IF V2<V1 THEN X=X+V2*DX:Y=Y+V2*DY:DY
=-DY:DRAW 1 TO X,Y:GO TO 200
180 X=X+V1*DX:Y=Y+V2*DY:DRAW 1 TO X,Y
190 X=XX:Y=YY:DX=-1:DY=-1:DD=DD+1:DRAW
1,X,Y:GO TO 210
200 IF XX=0 THEN XX=X:YY=Y:ELSE HH=1
205 IF X=XX AND Y=YY AND HH=1 AND DD=0 T
HEN END
210 IF DD=2 THEN END
220 GO TO 150
```

200–220 sorokban vizsgáljuk: abba lehet-e hagyni? 205 – sarok érintése helyett visszatér önmagába. 210 – kétszer volt sarokban.

A harmadik kérdéshez először próbáljuk meg jellemezni az ábrát telítettség szempontjából! A gyerekek ilyeneket fognak javasolni: számoljuk meg a pontokat, milyen távol vannak egymástól a párhuzamosok, hány metszéspont van stb. Javasoljuk azt nekik, figyeljék meg, hogy hányszor érinti az oldalakat a bolyongás, azaz egy számpár (X<sub>9</sub>, Y<sub>9</sub>) legyen a jellemző. Nos, ez vajon mitől függ, és hogyan? Talán lesz aki észreveszi, hogy az oldalakkal (A, B) van kapcsolatban a dolog, de hogy hogyan...

További segítség, ha az oldalakat INPUT-tal adjuk meg, és így már mi szabhatjuk meg, milyen téglalapon kísérletezzünk. Készítsünk táblázatot a tanulókkal, és talán már 4–5 próbálkozás után születik valamilyen hasznos megfigyelés.

**Mi az egzakt összefüggés? Telítettség szempontjából van-e jelentősége a kezdőpont megválasztásának? Érdemes elgondolkozni azon, hogyan jellemezhető például a metszéspontok száma, vagy a kivilágított pontok száma az eddigiekkel (A, B, X<sub>9</sub>, Y<sub>9</sub>). Ezekre is várunk válaszokat.**

**2. feladat.** Írjunk programot, mely tetszőleges  $x \rightarrow f(x)$  függvényt ábrázol, sőt a görbét „optimálisan” helyezi el a képernyőn! Számítalan megoldás ismeretes, közlünk egy nem túl bonyolult változatot:

```
5 DIM A(150)
10 DEFFN F(X)=1/((X-1)*(X+1)*(X+2)*X)
50 N=150:INPUT "INTERVALLUM (A,B)";A,B
70 DX=(B-A)/N:MA=FN(A):MI=MA
90 TRAP 300
100 FOR I=1 TO N
110 A(I)=FN(A+I*DX):IF A(I)>MA THEN MA=
A(I)
115 IF A(I)<MI THEN MI=A(I)
120 NEXT I
150 DY=190/(MA-MI)
200 GRAPHIC 3,1
207 DRAW 1,5,(A(1)-MI)*DY
210 FOR I=1 TO N
220 DRAW 1 TO I+5,(A(I)-MI)*DY
230 NEXT I
299 END
300 RESUME 120
```

10-ben adható meg a függvény egyenlete, míg a 20-as sorban az  $x \in [A, B]$  értelmezési tartományt jelölhetjük ki. Fontos, hogy az A pontban valóban legyen értelmezve a függvény – esetünkben A bármi lehet –2, –1, 0, 1 kivételével. A többi (belső) pontra van védelem a 90. sorban. 100–150 sorokban az Y szélsőértékei (MAX és MIN) segítségével beállítjuk a függőleges léptéket, majd ennek alapján ábrázoljuk a már tárolt pontokat (A(I)–k). DX és DY meghatározását automatikus normálásnak nevezzük.

**3. feladat.** Ha egy  $R = 100$  sugarú kört akarunk a képernyőre rajzolni, akkor használhatjuk a következő összefüggést is:

$$x = R \cdot \cos(t) \quad y = R \cdot \sin(t).$$

Az ilyen megadást a görbék paraméteres (implicit) egyenletének nevezzük. Elég annyit tudni róla, hogy a programban t paraméter lesz a ciklus változó, és a pontok x, y koordinátáit a fenti két összefüggés adja. Még arról kell gondoskodnunk, hogy a kör férjen el teljes egészében a TV-n, sőt minél „optimálisabban” helyezzük el.

```
10 DEFFN F(X)=100*COS(T):DEFFN Y(T)=100*SI
N(T)
50 N=150:A=0:B=2*pi
70 DT=(B-A)/N
200 GRAPHIC 1,1
207 DRAW 1,FN(X)+100,FN(Y)+100
210 FOR T=A TO B STEP DT
220 DRAW 1 TO 100+FN(X),100+FN(Y)
230 NEXT T
```

Sok érdekes ábrát nyerhetünk, ha gondoskodunk a programban a megfelelő elhelyezésről, az automatikus normálásról (az X és Y léptékek kiválasztása). Paraméteres megadásnál ez egy kicsit bonyolultabb, (és lassabb is) mint az  $x \rightarrow f(x)$  esetben, de némi töprengés után adódik:

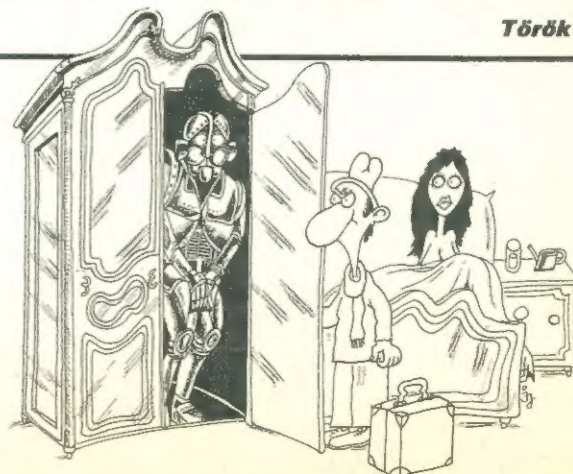
```
5 DIM X(200),Y(200)
10 DEFFN F(X)=COS(T):DEFFN Y(T)=SIN(T)
50 N=200:INPUT "INTERVALLUM (A,B)";A,B
70 DT=(B-A)/N:XA=FN(A):XF=XA:YA=FN(A):
YF=YA
90 TRAP 300
100 FOR T=A TO B STEP DT
110 X(I)=FN(X):IF X(I)>XF THEN XF=X(I)
115 IF X(I)<XA THEN XA=X(I)
120 Y(I)=FN(Y):IF Y(I)>YF THEN YF=Y(I)
125 IF Y(I)<YA THEN YA=Y(I)
130 I=I+1:NEXT T
150 DX=300/(XF-XA):DY=190/(YF-YA)
200 GRAPHIC 1,1
207 DRAW 1,(X(1)-XA)*DX,(Y(1)-YA)*DY
210 FOR I=2 TO N-1
220 DRAW 1 TO (X(I)-XA)*DX,(Y(I)-YA)*DY
230 NEXT I
```

Ez a programunk már túlságosan is önálló: például nem hajlandó kört rajzolni. Kárpótlásul néhány érdekesebb egyenletet adunk meg, lehet gyönyörködni.

$x = a \cdot t / (1 + t^2)$	$y = a \cdot t^2 / (1 + t^2)$	– Descartes-levél
$x = a \cdot t^2 / (1 + t^2)$	$y = a \cdot t^2 / (1 + t^2)$	– ciszoid
$x = \cos(t) \cdot (1 + \cos(t))$	$y = \sin(t) \cdot (1 + \cos(t))$	– kardioid
$x = a \cdot (t - b \cdot \sin(t))$	$y = a \cdot (1 - b \cdot \cos(t))$	– ciklois
$x = a \cdot \cos^3(t)$	$y = a \cdot \sin^3(t)$	– asztrois
$x = \cos(a \cdot t)$	$y = \sin(b \cdot t)$	– Lissajoux-görbe

Utolsó egyenleteink pontosan olyan problematikához vezetnek, mint amit az 1. feladatnál gondoltunk végig. Az ábra bonyolultságát lényegében ugyanúgy befolyásolják most az a és b együtthatók, mint annak idején a téglalap oldalai. További analógia, hogy mindkét ábra hengerre rajzolva képzelhető el igazán: például egy jobb oldali (felső) határpont előbb-utóbb megjelenik baloldalt (alul) a téglalapon. Másként fogalmazva: az 1. feladatbeli bolyongás tulajdonképpen egy henger palástján történik, majd alkalmas helyen felvágva a palástot, és kiterítve adódik a kívánt téglalap. A Lissajoux görbék III. osztályban fizikában szerepelnek.

**Török Turul**







# K Ö N Y V M O L Y

**Meggyesházi Péter-Pintér József: ZX Spectrum haladóknak** – Műszaki Könyvkiadó, 199. o., 78 Ft.

(A szerzők röviden ismertetik a Spectrum sajátosságait, a Z80 assembly használatát, a ROM-programot, valamint az Interface 1-et és a Microdrive-ot. Ezután gazdag programkatalógust, valamint egy jól használható rutinyűjtemény-listát közölnek.)

**C16 és Plus/4 programozói útmutató** – Novotrade, 86. o., 129 Ft

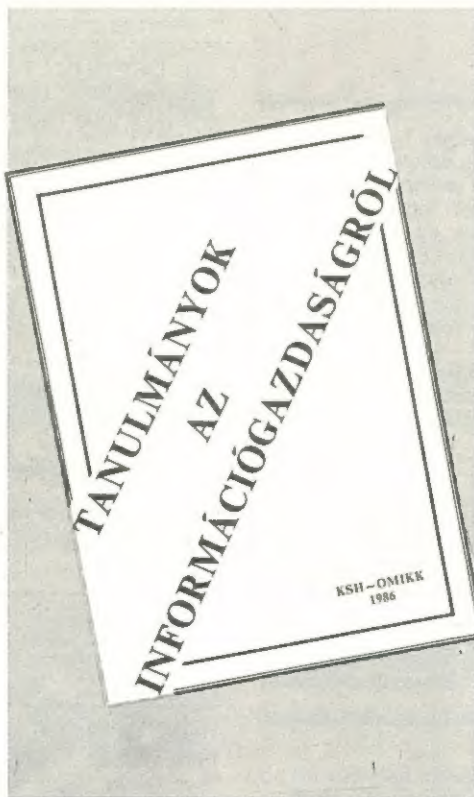
(A programozási kézikönyv folytatása a BASIC alapismeretek áttekintése után a gépi kódú programozásba nyújt betekintést. A kötetet számos, jól használható függelék egészíti ki.)

**Grohmann-Eichler: A 68 000-es mikroprocesszor** – DATA BECKER – Novotrade, 368. o., 349 Ft.

(A könyv a Motorola mikroprocesszor-családot mutatja be: ismerteti kifejlesztésének történetét, felépítését, input-output jellemzőit, utasításkészletét. A kötet szerzői kitérnek arra is, hogy a 68 000-es mennyiben tér el a többi 16 bites processzortól.)

**Sasse: Compiler** – DATA BECKER – Novotrade, 350 o., 298 Ft.

(A szerző a programnyelveket fordító rendszerek alapismereteibe vezeti be az olvasót, egy erre a célra kifejlesztett programnyelv segítségével. Megismerhetők a lexikális, a szintaktikai és a szemantikai elemzés szabályai, valamint az assembler néhány sajátossága.)



nevezhető, hiszen előzmények nélküli vállalkozás volt. Szerte a világban tíz éve hatalmas vitát folytatnak a szakemberek az informatika gazdasági szerepéről, hatásáról – erről ezidáig itthon nem vettünk tudomást. Emiatt megérthetők és elfogadhatók a kötet gyermekbetegségei. Nem hibáztathatók a népes szerzőgárda tagjai amiatt, hogy szinte mindegyikük az alapfogalmaktól indul el, azokat magyarázza részletesen, és ennek alapján építi fel saját mondanivalóját, – hiszen egy konferencián ez természetes. Az sem róható fel nekik, hogy az egyes fogalmak definíciói, meghatározásai nem mindig egyértelműek, nem mindig letisztultak – elvégre egy most kialakulóban lévő tudományág részei. A nyelvezet is körülményes néhol, nehezen átrágható.

A szerzők saját szakterületüknek megfelelően fejtik ki tapasztalataikat, kutatási eredményeiket az információgazdaságról. Így a legkülönbözőbb szakágak képviselőinek véleményét olvashatjuk, a politológiától a térképészig. Mindegyikük megpróbálja adap-

tálni saját szakmájában a külföldi eredményeket. Ez érthető álláspont – de a könyvből nem sugárzik egységes szemléletmód. Heterogén a kötet, és a szerkesztő nagy érdeme, hogy az említett hibák ellenére egyhuzamban végigolvasható.

Lemaradásunk az informatikában a fejlett országokhoz képest hatalmas. Hogy mennyi, arról eltér a szerzők véleménye (10–20–30 év), de az egyértelműen kiderül, hogy néhány mutató szerint a fejlődő országokkal vagyunk egy szinten. A közgazdászok számára hatalmas felkiáltójele lehet ez a kötet, hiszen a reformelképzelésekben is csak huszadrangú tényezőként kap szerepet az informatika – mostanáig nem jöttünk rá, hogy ez mindenhol húzóíparág.

Talán napjainkban ébrednek rá mind több helyen az informatikai kutatások szerepére – és ez is aláhúzza a kötet fontosságát –, hiszen egyre több vállalat, intézmény kér fel szakembereket annak vizsgálatára, hogy egy-egy tervezett informatikai fejlesztésnek milyen eredményei, hatásai várhatóak. S a hatások alatt nem csak a gazdasági eredményeket kell értenünk, hanem azt is, hogy a fejlesztés nyilvánvalóan befolyásolja az adott vállalat szerkezetét is. Itt kell megemlítenünk a kötet egy másik fogyatékoságát: a tanulmányok szerzőinek legtöbbször erősen technokrata szemszögből elemzi a bekövetkezett és a várható változásokat. Hiányzik a társadalomtudományi, a szociológiai megközelítés, pedig minden forradalmi jelentőségű ipari megújódás esetében nyilvánvaló, hogy a technológia nem előidézője a forradalomnak, hanem csupán (?) kiteljesedése. Az összeállításban olvashatunk ugyan erre vonatkozó utalásokat, de annak elemzésére, hogy az informatika magyarországi elterjedésének mik a társadalmi előfeltételei, nem vállalkozik egyik szerző sem. A kötet mindenesetre kísérletet tett az információgazdaság fogalmának tisztázására. Nem biztos, hogy a tanulmányokból leszűrhető definíció mindenben helytálló, de konzisztens, ellentmondásmentes egészet képez. Levonható az a következtetés is, hogy égető szükség van országos, átfogó információs politika kidolgozására. Így a könyv nemcsak a közgazdászok, a gazdasági vezetők számára lehet fontos és hasznos olvasmány, hanem a társadalomtudósok, a politikával foglalkozó szakemberek számára is – sőt oktatási célokra is ajánlható.

Az első kísérlet nyilvánvalóan további kutatásoknak kell követniük, de már ez a kötet is figyelemfelkeltő és gondolatébresztő.

**Tallér József**

## Tanulmányok az információgazdaságról

**(Szerkesztő: Szabó József)** – KSH-OMIKK, 234. o., 260 Ft.

Előző számunkban bemutattuk a Statisztikai Kiadó kötetét, mely a francia informatikai forradalommal, illetve annak gazdasági, társadalmi hatásaival foglalkozik.

Az informatika kihívása ugyanúgy érezhető Magyarországon is, mint bármely más országban – ezért szervezett 1985-ben a Központi Statisztikai Hivatal szemináriumot az információgazdaságról. A szemináriumra készült előadásokból ad válogatást a most megjelent kötet.

A megrendezett konferencia frontáttörésnek



Az elmúlt nyáron az NSZK-ban vettem egy új sorozatú C64-et. Az eladók elmondták, hogy a régi típustól eltérő vonás az úgynevezett GEOS megléte. Mi az a GEOS? Mire lehet használni? Hogy lehet működésbe hozni?

Kresz Gábor

Boly Szabadság u. 49.

A GEOS egy új rendszerprogram a C64-hez. Ön kellett, hogy kapjon a gépéhez egy lemezt és egy kis használatú útmutatót. Ezen a lemezen van a GEOS, ha ezt betölti egy alapvetően más rendszerben használhatja a gépet joystick vagy egér segítségével. A lemezén a rendszerben írott két komolyabb felhasználói programot is talál: a GEOPAINT-et és a GEOWRITE-ot. Előbbi egy kitűnő rajzoló, utóbbi egy szövegszerkesztő-féleség. Hogy ezeket a programokat és egyáltalán magát a rendszert hogyan kell használni, azt a használati útmutatóból tudhatja meg. Mivel ezt ön nyilván németül kapta meg a gépéhez, föl hívjuk a figyelmét, hogy az Országos Commodore Egyesület C=újságjában a januári számmal kezdődően sorozatban közlik a GEOS rövidített magyar nyelvű használati útmutatóját. (A lapot az egyesület tagjai kaphatják, s hogy hogy lehet ön is az egyesület tagja, ezt többek kérésére az alábbiakban közöljük.)

## COMMODORE EGYESÜLET

Mint hogy többen kérdezik tőlünk is, hogy hogyan lehet hozzájutni a Commodore Egyesület 36 oldalas havonta megjelenő Commodore újságjához, ezúton közöljük, hogy ehhez be kell lépni az Egyesületbe. Évi 1264 forint a tagdíj, ezért havonta megküldik az újságot, benne egy vagy két 50 forintos vásárlási tikkettel. (Igy tehát a tagdíj nagyobbik része visszatarú.) A tagdíjat be lehet fizetni egyszerű postai pénzküldő csekkben az alábbi címet kell ráírni (nemcsak a címzett rubrikába, hanem a csekk hátulján lévő közlemények helyére is): Commodore Egyesület - OTP Budapest XIII., Visegrádi u. 7/b. MNB 217-98292, OTP 565-3610

Többen kérdezték tőlünk, hogy hogyan lehetne hozzájutni lapunk decemberi számában csak egy fotó erejéig bemutatott Szarka György és Tihor Miklós által készített, s legjobbnak minősített Vadász és a nyúl című játékprogramhoz. Nos régi szokásunk, hogy olvasóinkat, pláne meg programozó barátainkat szívesen összehozzuk egymással. (Láthatják, hogy hacsak külön nem kéri valaki ennek az ellenkezőjét, rendre közöljük szerzőink lakáscímét is.) Ime tehát Tihor Miklós címe is. De a válasza váróknak legyenek türelemmel, mert szegény e pillanatban előfelvételi lévén sorkatonai szolgálatát tölti. A cím: Tihor Miklós - 1181 Budapest, Városház u. 31.



# PLUS/4 NYERŐ

## PLUS 4 NYERŐ 2. FELADATÁNAK MEGOLDÁSA

Helyhiány miatt csak egy vázlatos megoldást közlünk. Könnyű volt észrevenni, hogy a tulajdonképpeni stratégia a H mátrixban van eltárolva. A H mátrix így néz ki:

A H(,) tömb DIM H(6,33)																																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	
0																																			
1		0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	0	0	2	0	1	1	1	1	0	0	0	1	0	0	0	0	1	0	
2		0	6	2	6	4	1	6	5	0	5	2	5	2	1	4	5	4	2	5	5	1	5	5	6	4	4	3	4	5	4	6	5	0	
3		2	1	2	1	2	2	2	2	1	2	1	2	2	2	2	2	2	1	2	2	2	1	2	1	2	2	1	2	2	2	2	2	2	
4		1	3	9	5	11	7	24	1	0	11	33	20	13	15	0	16	5	19	27	20	26	22	17	30	22	31	3	28	29	3	11	9	0	13
5		1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	4	0	0	2	0	3	0	2	0	0	0	0	0	0	0	1	0	0
6		1	0	3	0	5	0	7	0	1	0	3	0	3	0	0	0	2	0	0	4	0	5	0	6	0	0	0	0	0	0	0	7	0	0

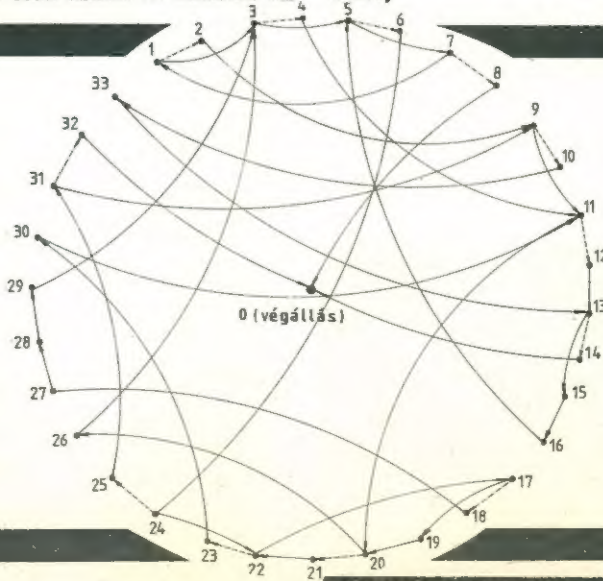
Az oszlopszám az állás sorszámát jelenti, a következőket figyelembe véve: Az állásokat első lépésben csak a vadászok helyzete alapján különböztetjük meg. Két állást azonosnak tekintünk, ha a 4 vadász ugyanabban a konfigurációban, vagy annak tükörképében van. Az „ugyanabban a konfigurációban” magában foglalja a bal szélről való távolságokat, de az egész konfigurációnak a tábla aljától vett távolságát nem. Így, ha a 4 vadász alulról a 2. sorban van, az tükörképe a kiinduló állásnak, ha a 3-ban, az azonos vele.

Egy állás alállásának azt nevezzük, ha a nyúl pont azon a pozíción áll, ahová valamelyik vadász éppen lépni szeretne. A nyúl többi helyzete szerint nem különböztetjük meg. Egy menet közben kialakuló vadász-konfigurációt úgy jellemezhetünk, hogy megadjuk melyik alap-konfigurációnak felel meg, annak tükörképe, vagy nem, s hogy a legalsó vadász alulról hanyadik sorban áll.

A H mátrix értelmezése: 1-2. sor annak a vadásznak a relatív koordinátái, mellyel lépni szeretne a gép. Az első koordinátához hozzáadva a legalsó vadász helyének sor-koordinátáját, a másodikat pedig 7-ből kivonva, ha tükörképről van szó, megkapjuk az illető vadász abszolút koordinátáit (mindkettő: 0-tól 7-ig terjedhet). A 3. sor a kívánt lépés irányát adja meg: 1 - jobbra, 2 - balra tükörkép állás esetén persze fordítva.

A 4. sor adja meg, hogy ezt lépve hányas állásba jutunk, ill. 0, ha ezt lépve a vadász nyer, az 5. és 6. pedig azt, hogy hová kerül a vadász, de ezt csak abban az esetben, ha

lehetséges, hogy a nyúl ott áll. Ebben az esetben lépéskor a megfelelő alállást kell tekinteni, melynek sorszáma mindig 1-gyel nagyobb az állás sorszámánál. Észrevétel: az alállásoknak nincs alállásuk, hiszen ide csak úgy kerülhetünk, ha tudjuk, hogy a nyúl az eredeti állás úgynevezett érzékeny pontján áll, s az alállásban mindig ennek megfelelően máshová próbálunk lépni. A játék az 1-es állásból indul, s az 1. ábra szerint folthat le: a rendes nyíl jelenti azt, hogy azt léptük, amit szerettünk volna, a szaggatott nyíl azt, hogy az érzékeny ponton áll a nyúl, így az alállás szerint kell lépni. Azt kell még megneznünk, hogy miért nincs minden állásnak alállása? 9 állásnak nincs alállása, ezek: 15, 16, 19, 26, 27, 28, 29, 30, 33. A magyarázat az, hogy ezek úgy alakulhatnak csak ki, ha néhány lépéssel hamarabb a nyúl az akkori állás érzékeny pontján állt, így kiszámítható, hogy az azóta eltelt lépései alatt milyen messze juthatott. Ha az állás érzékeny pontját semmiképp sem érthette el, akkor nincs szükség alállásra. Ez mind a 9 esetben ellenőrizhető az 1. ábra segítségével, csak fel kell rajzolni az egyes állás-sorszámokhoz tartozó ábrákat. Ezt a 21 ábrát hely híján nem tudjuk közölni. Az eddigieket elfogadva a stratégia működőképessége könnyen látható, ugyanis pl. az előbb említett módon minden álláshoz hozzárendelhető a nyúl összes lehetséges helyzete, s látható, hogy a nyúl sose tud a vadászok mögé kerülni. Így a játék csakis a vadászok győzelmevel érhet véget (csak 8 sor van!)





### GÉPNYERŐ ÉRTÉKELESE

A februári Gépnyszerősorsoláson a **Commodore Plusz 4-es** gépet **Szilvási Margit-nak** (Budapest) húztuk ki. **1 doboz disket** nyert: **Kruzslicz Ferenc**, Tótkomlós, **1000 forintos vásárlási utalványt Szabó Zoltán** Tíbor, Budapest, és **500-500 forintos utalványt:**

**Kozma Benedek**, Budapest  
**Tinkó Attila**, Budapest  
**Kovács Gábor**, Vác  
**Kurusa Árpád**, Szeged  
**Horváth Péter**, Budapest  
**Peták Tamás**, Szolnok  
**Földvári Csongor**, Budapest

### A HARMADGÉPNYERŐ ÉRTÉKELESE

Mindhárom feladatra csak 20 pályázónk küldött be megoldást. Az első feladat elég könnyűnek bizonyult, bár 1-2 részfeladatot sokan elrontottak. A második sem volt annyira nehéz, mint gondoltuk, a mienknél egyszerűbb megoldások is érkeztek. Többen nem vették észre, hogy a 17-es feladat visszavezethető-e a 16-osra, s így külön hosszan bizonyították. A harmadik feladat rázósabbnak bizonyult. Két fő hiba: többen nem végezték el a lépésszámbecslést, és többen kódolásnál is komoly algoritmus lépéseket végeztek szegény kódolással, ráadásul a kódoló algoritmusának lépésszámát általában nem becsülték meg.

**A potenciális nyertesek:**

**2 cédulával:** **Földvári Csongor**, Budapest; **Szabó Zoltán**, Budapest; **Biró Gábor**, Kisvárd; **Voronai László**, Budapest; **Morvai Lajos**, Budapest; **Solymosi György**, Budapest; **Peták Tamás**, Szolnok

**1 cédulával:** **Fekete Zoltán**, Szolnok; **Veress Zoltán**, Újfehértó; **Almássy Zoltán**, Miskolc  
 A sorsolásra 1987. április 18-án 10 órakor kerül sor az **Almássy téri Szabadidő Központ** Compánia számítógépes műhelyében.

**Figyelem!** Múlt havi számunkban tévesen jelent meg a Quatroplus nyerő sorsolási időpontja. Azt is ugyanebben az időpontban sorsoljuk!

Amint azt már a múlt hónapban beharangoztuk, három hónapos pályázatot indítunk, amelynek első díját a Skála Computer S hálózata ajánlotta föl. A nyereményért három olyan feladatot kell megoldani, amelyek ugyan apró kis programok, de megoldhatók és értékelhetők gép nélkül is.

Nagyon kérjük az olvasókat, hogy megoldásaikat ne kazettán, lemezen küldjék be! Egyszerűen csak írják le egy papírra.

# COMPUTER-S ATARI NYERŐ

## 1. FELADAT

Az viszont értelemeszerű, hogy a pályázóknak legalább alapfokon ismerni kell a BASIC nyelvet.

Az első feladat: egy gép (pl. CMIN8 a neve) BASIC-je igen kevés utasítást ismer:

– A PRINT, LET, INPUT, DIM és END utasítások a legtöbb BASIC-hez hasonlóan működnek.

– Ismeri a GOTO K utasítást, ahol K tetszőleges aritmetikai kifejezés. Ha ilyen utasításhoz ér, kiszámítja K értékét, s ha van K sorszámú sor, akkor oda ugrik, különben hibát jelez.

– Ismeri a +, -, \*, /, ↑ műveleteket és a <, >, =, <=, >=, <> relációkat, melyeket műveletnek tekint, s az eredmény -1, ha a reláció fennáll, s 0, ha nem. Precedencia (elsőbbségi) sorrend:

1: ↑    2: \*/    3: +, -

4: <, >, =, <=, >=, <>

Azonos „erősségű” műveleteket balról jobbra végez el.

Zárójelezés lehetséges.

– Ismeri a BASIC alapfüggvényeit:

ABS, ATN, COS, EXP, INT, LOG, SGN, SIN, SQR

– Tud kezelni stringeket, stringek között értelmes a + művelet (konkatenáció) és az =, <> relációk. Ismeri az INKEY\$ függvényt is.

– 1 sorba 1 utasítás írható!

Írjunk ebben a BASIC-ben (tehát csak a felsorolt utasítások használhatók) egy programot, mely a következőket tudja:

Bekérdezi N értékét, ellenőrzi, hogy pozitív egész szám-e. Ha nem, akkor újra kérdezi.

Ezután sorban bekérdezi N db számot, majd ezeket sorbarendezve (növekvő sorrendben) kiírja.

Ezután vár egy tetszőleges billentyű leütésére.

(Aki az INKEY\$-t nem ismerné, leírjuk hagyományos BASIC-ben ennek megvalósítását:

100 Q\$=INKEY\$

101 IF Q\$="" THEN GOTO 100.)

– Ha ez a billentyű K, I vagy SPACE, akkor megvizsgálja, hogy a sorbarendezéskor kapott legkisebb szám pozitív 10-nél kisebb egész szám-e, (ha nem, ezt szöveggel jelzi!) ha igen, akkor kiírja a számot betűvel két nyelven, s megáll.

– Ha a leütött billentyű más, elbúcsúzik és megáll.

A programokat papíron kérjük beküldeni, lehetőleg rövid magyarázattal. A rövidebb, elegánsabb programok több pontot kapnak!

